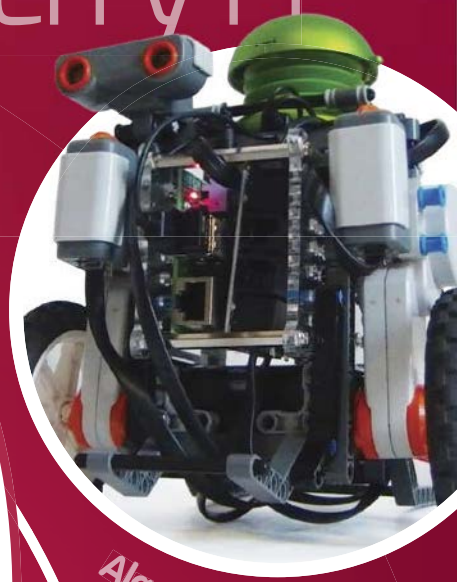
RasPi

DESIGN
BUILD
CODE

19

Get hands-on with your Raspberry Pi

Digital camera conversion



Alarm clock robot

20

Fireball pinball

RetroNES

Raspberry Pi Projects

Top gadgets hacked by **you**

Ben Heck's hacks

Resurrected radio

Plus Portable arcade, spaceship bedroom & more

Welcome

Raspberry Pi coders and makers are some of the most inspiring people around – this is a *great* community of really creative individuals and groups. It's always inspiring to see what you are all up to, so for this issue we've shared some excellent creations from among the myriad that we keep tabs on – hopefully you'll find something that inspires you as much as it did us! We love a good hacking project here at **RasPi**, so we've picked 20 of our favourites – from the chap who built a spaceship and mission control desk into his sons' bedrooms through to element14's hacker extraordinaire Ben Heck, we have the inside story from each maker. Plus, you can finish off your J.A.R.V.I.S. software this issue!

Gavin Thomas

Editor

Get inspired

Discover the RasPi community's best projects

Expert advice

Got a question? Get in touch and we'll give you a hand

Easy-to-follow guides

Learn to make and code gadgets with Raspberry Pi

From the makers of

LinuxUser & Developer

Join the conversation at...

@linuxusermag | Linux User & Developer | RasPi@imagine-publishing.co.uk

Contents

20 Raspberry Pi projects

Get the inside story on the greatest Raspberry Pi hardware hacks

There are now around six million Raspberry Pi models out in the wild, and some of the things that you, the Raspberry Pi community, have made with them truly are wild. From elegantly crafted scripts that chain together a series of web services to homebrew Rube Goldberg machines, they are as creative as they are diverse. And through the crowd of new projects bubbling up online every day, if there's one word that's guaranteed to get everyone's attention then it's the word 'hack'.

But what exactly is a hack? Well, for the purposes of this feature, we decided that a hack has to have some sort of hardware base. It's the kind of project where you take one device and, with a little Raspberry Pi magic, transform it into something wholly new and original, extending the functionality of the base hardware in ways that were never anticipated. It riffs off the traditional definition of the hacker, which grew out of MIT's Tech Model Railroad Club. These are the projects that get us excited and make us want to learn more about electronics, engineering and programming.

Over the next few pages we're going to introduce you to some of the greatest Raspberry Pi hacks that we've ever discovered. Projects where vintage hardware has been torn apart and the components repurposed into something amazing, or where the hardware has been puzzled over, fiddled with, then brought back to life after twenty years spent languishing in a garage. These hacks inspire us, with each maker striking the right balance between passion, skill and virtuosity, and we hope they inspire you too. Read on as we hear how you can launch a satellite from a bedroom spaceship, transform an analogue camera into a digital one, make a classic Apple Pi and more.

01

Portable Pi Arcade

Love old arcade games? With Ben Heck's hack, you can play them all on a single hand-built, hand-held console

Ben Heck has built two versions of the Portable Pi Arcade. The first was the original Portable Raspberry Pi project (**https://www.youtube.com/watch?v=dUZjzQuTNX4**), where he hacked the Pi to reduce its size and opened up a USB game controller to extract the circuits. With a new assembly in place, he 3D-printed a custom-designed case, put the new device together, and booted up MAME (the Multiple Arcade Machine Emulator) to play old-school games.

Left Install MAME and you will never run out of arcade games to play

MAKER PROFILE

Ben Heck

Master hacker, creator of The Ben Heck Show

Ben Heck is an online sensation and a pillar of the maker community, putting out amazing how-to videos for games console hacks and all kinds of different Pi projects. He's done it all on element14's *The Ben Heck Show*.

Find out more...
benheck.com

Left Watch the video to see how Ben builds and tests the custom circuit board

His revival of this earlier project was even more home-made. For the Raspberry Pi MAME portable gaming device (**https://www.youtube.com/watch?v=zrEj1aQRbpw**), Ben made a circuit board from scratch, fitting the components into a new 3D-printed case that, rather than resembling a Game Gear, looks pretty close to a Game Boy.

We asked Ben to take us through the original version of his project: "My first portable Pi project was a small, battery-powered unit for gaming," he begins. "It had a single USB port for Wi-Fi or external storage and we featured it back on season three of the show. The screen came from a cheap NTSC LCD screen that Amazon sells for use as a car's backup camera. The buttons I laser-cut myself and the case was 3D-printed." And in terms of physical modifications to the Pi? "Mostly I removed the taller through-hole components," he replies, "and attached the Teensy HID (used for controllers) directly to it. I also moved the secondary USB port."

"The screen came from a cheap NTSC LCD screen that Amazon sells for use as a car's backup camera. The buttons I laser-cut myself"

As you can see below, the case is very well made. "I did the initial layout in Adobe Illustrator, for the laser-cut portions," explains Ben, "then transferred the whole design to Autodesk 123D to create a 3D-printable file. Hand-writing the buttons for the controls was the most challenging part of this project. It was the most time-intensive part and required a lot of precision and attention to detail."

Ben is no stranger to taking apart consoles and controllers for his Pi hacks – but he also makes one-handed accessibility controllers. "In addition to all of the other projects and hacks, we modify gaming controllers for people who have difficulty using existing ones," Ben tells us. "On the show we've featured a few of them – Xbox One, PS4, even the Wii. Now we build these controllers by request and they can be ordered off my website, though we only do Xbox 360/Xbox One controllers as those use PCBs throughout (instead of silk screen circuits like the PS4). Recently I trained Felix (an assistant on element14's The Ben Heck Show) on how to do it, so he's been helping and working on them in his spare time as well."

Heck's hacks

Pi Point and Shoot: A Raspberry Pi camera module, PiTFT from Adafruit, PlayStation 3 controller battery and additional parts were all made into a point-and-shoot camera.

Pi Retro Computer: A tribute to the BBC Microcomputer from the 1980s, Ben mounted a Raspberry Pi to a self-made wooden case, HDMI port, on/off switch and USB hub for an 'old-school' feel computer and carrying case.

Handheld Pi Console: Ben hacked a Raspberry Pi single board computer to make it smaller. Combined with a composite LCD wireless keyboard, lithium battery power source and USB joystick, he created a handheld console.

Left Combining 3D-printing and laser-cutting leads to great bespoke designs

02

Camera Pi

Power up a regular point-and-shoot DSLR camera so that it can send fresh photos straight to your tablet

Dave Hunt is an excellent photographer and shared his Camera Pi invention with us. "I needed to transmit photos to an iPad as they were taken," explains Dave, "but the commercial solutions were £500. I had a broken battery grip big enough to fit my Raspberry Pi and a battery, so it went from there.

"The battery grip holds two batteries. Once I'd stripped out the battery compartment, I set about filing down all the mounting holes inside the grip so I could get the Raspberry Pi inside.

"The next task was to fit a camera battery and DC-DC converter inside. I was able to use part of the removed internals of the grip, and before long I had a slot to insert a camera battery into. It's capable of powering the Pi for about four hours.

"Making it wireless was a case of plugging in a USB Wi-Fi adapter. A few lines of Perl later and I was able to poll the camera with gphoto2, pull the new files off and send them via FTP to ShutterSnitch on my iPad."

Read up on the full build process and check out Dave's video at **http://bit.ly/1BxEMbC**.

MAKER PROFILE

Dave Hunt

Photographer and maker

David has been making projects for the Raspberry Pi since the early days.

Find out more...

davidhunt.ie

03

Pi Telephone

Revive a ringing phone with C# circuit wizardry and voltage manipulation

Stuart Johnson is bringing a classic GPO 746 handset back to life, and while the project wasn't complete at the timing of writing, he has finished the lion's share of it. "I took out the main circuit board inside the phone and squeezed the Raspberry Pi in there," says Stuart. "I was then faced with two challenges – the biggest one was getting the bell to ring. I found a solution by raising the voltage to 19 volts and dropping it down to 5 for the Ras Pi using a very small DC-DC converter (the OKI-78SR), with the rest then being used for the bell. I was surprised by how well it worked.

"The bell is using one of the I/O ports, and there's an available C# library (raspberry-sharp-io) which lets you monitor and control those ports. So I linked one of the I/Os to the pulse dial and connected another to a relay using transistors. Then with the software I put in a timer to measure the pulse clicks. I managed to write some code to time those pulse clicks and determine the number dialled."

MAKER PROFILE

Stuart Johnson
Managing director

Stuart runs Logic Ethos, an IT company in Southampton providing network services and cloud computing to developers.

Find out more...
logicethos.com

Above
Remember these things? It's easier than you think to resurrect one!

04

Car Computer

Ours was good, but Derek Knaggs really has built the real deal straight into his glove box

Remember the sat-nav we built a couple of issues back? Well, Derek Knaggs already beat us to it, and he's embedded the display in his dashboard and extended the setup to include screens for the rear passenger seats too. "I removed the DVD player, which was a standard Ford head unit," explains Derek, "and then purchased a head unit from Xtrons. It's designed for the Ford Focus so it was a straight swap. The Xtrons radio has an S-Video input and that goes into the radio, so the Raspberry Pi displays as Auxiliary Input.

There's two Auxiliary Outputs on the radio, so the Raspberry Pi sends a video to the main radio which then sends it back out to the screens in the passenger seats. What I've done is put in a device – like a VGA adaptor: it takes one input and puts seven out – that gives me the ability to have the Raspberry Pi running to the back screens on their own, so the radio can then control itself. I can have my kids watching movies at the back with the Raspberry Pi using an audio splitter (they've got headphones on), and we can be at the front using the controls, like the sat-nav."

MAKER PROFILE

Derek Knaggs
Managing director

Derek Knaggs runs Flamelily IT, an IT supply and support company, and is studying Computing at the University of Worcester.

Find out more...
flamelily.co.uk

05

RetroNES

Now you're playing with power. Raspberry Pi power to be exact, situated inside an old game console

"It all started when my wife asked me what I wanted for Christmas", said Chris over email. "I had absolutely no idea but I had been wanting to mess around with a Raspberry Pi since it came out, so she got me a starter kit.

"While waiting for Christmas I started narrowing down ideas and found the RetroPie project. I thought that I would just install that, load some ROMs and call it good, then I remembered that I had some old NES and SNES controllers in storage. I went to get them and found my old childhood NES console along with the controllers. Once I got the NES

Left It's difficult to see, but there are some difference to this NES compared to an original

MAKER PROFILE

Chris Crowder

Programmer and database administrator

Working in the car industry, Chris develops manufacturing systems for production floor systems using .Net and SQL. In his down time, he likes to play videogames and tabletop games, but was previously limited to his PC for the former.

Find out more...

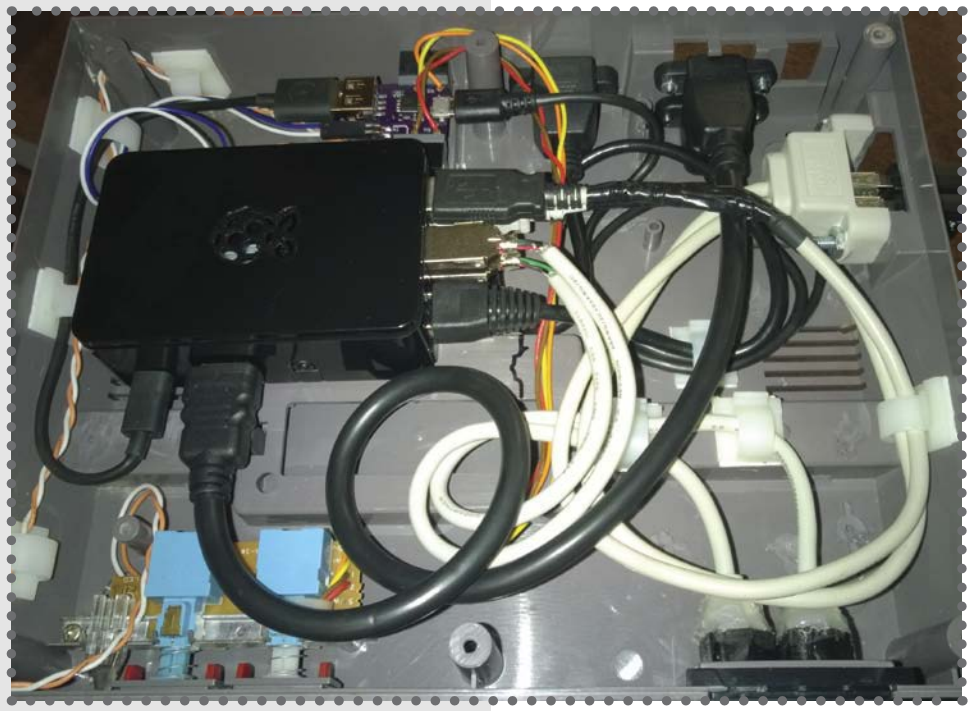
imgur.com/a/KPi2n?gallery

back in the house and started looking at it, I found that almost all of the internals were damaged due to insects and moisture. All of the connectors were corroded and some of the boards had traces that were peeling. That is when I decided that I would use the Raspberry Pi to 'resurrect' the NES."

Chris completely gutted the case and replaced the insides with a Raspberry Pi, hooking up I/O ports to the original connectors for the controllers and the AV cables and such. What's it like taking on a project with one of the most revered consoles in videogame history?

"It was a little intimidating at first as I wanted to make sure that this project looked and felt like an NES but with more flexibility. The biggest issue I ran into was that I wanted it to be able to work like an NES, meaning that if someone wants to play a game then they just turn it on, select a game and then they are playing. When they are finished all they have to do is press the power button to turn the console off. We can't do that easily with a Raspberry Pi since there is no ATX-style power switch. I was able to solve this issue with a Mausberry Circuit and a Python script. When the power button is pressed it communicates with the Pi via a GPIO connection and it runs the shutdown command. Once the Pi is shut down, the circuit cuts the power to the Pi."

The final product works great, with Chris reporting he can play on Atari, Sega and Nintendo games just fine. He's now looking to upgrade it with a Raspberry Pi 2 and increase the number of games he can play.



Right Everything is packed inside the original case, without needing to open it up to use it

Refitting a NES

Do you fancy bringing your old console back to gloriously pixelated life? Thought so. In that case, you'll be needing this list of the key components that Chris used to repair and revive his childhood NES console:

A broken NES console

Replacement NES Door

Canakit Starter B+

Panel Mount Ethernet, HDMI and USB cables

USB A Male Connectors 10pk

SNES USB Controllers

Anker 13000 mAh 3 watt Battery

Mausberry Circuit

3-volt LED

06

Mission Control Desk

Astronaut training begins early in Jeff Highsmith's home, with his sons running launches from their bedrooms

Jeff Highsmith is probably the best Dad in the world. Not content to just build his son a desk for his room, he modified it so that space adventures can start with the push of a button.

"My eldest son was starting kindergarten and he needed a desk to do his homework on," Jeff tells us, "and since I like to build stuff I thought I would make him a desk rather than buy one, and I was thinking, 'What would make

MAKER PROFILE

Jeff Highsmith

Tinkerer extraordinaire

Jeff Highsmith loves to make new and novel things. The medium isn't important, and he enjoys scrounging for materials and making do with what's at hand.

Find out more...

jeffhighsmith.com

Left The Mission Control desk groups the various functions into 'station' panels

Left Outside of playtime, this is just an ordinary homework desk. Almost

a really awesome desk?' Well, having lots of buttons and switches like a mission control desk! Carpentry-wise it was pretty simple to build.

"The Raspberry Pi is up in the front-centre behind a piece of flexiglass, next to the Arduino that takes care of reading the inputs. The Pi handles all the sounds and the logic – the gameplay aspect. That was my first Python experience and it was pretty good.

"The desk has got several modular panels and each has a different function. So in the real mission control at NASA there's a desk that controls the retro stage, and for this desk I made a retro booster panel and put a bunch of rocket noises on it. There's a capcom (capsule communicator) panel, so you put on a little headset and you can talk to the astronaut that is in the spaceship in the other room.

"There are a couple of panels that have numerical displays: one reads out some attitude numbers, like

"There's a capcom panel, so you put on a little headset and you can talk to the astronaut that is in the spaceship in the other room"

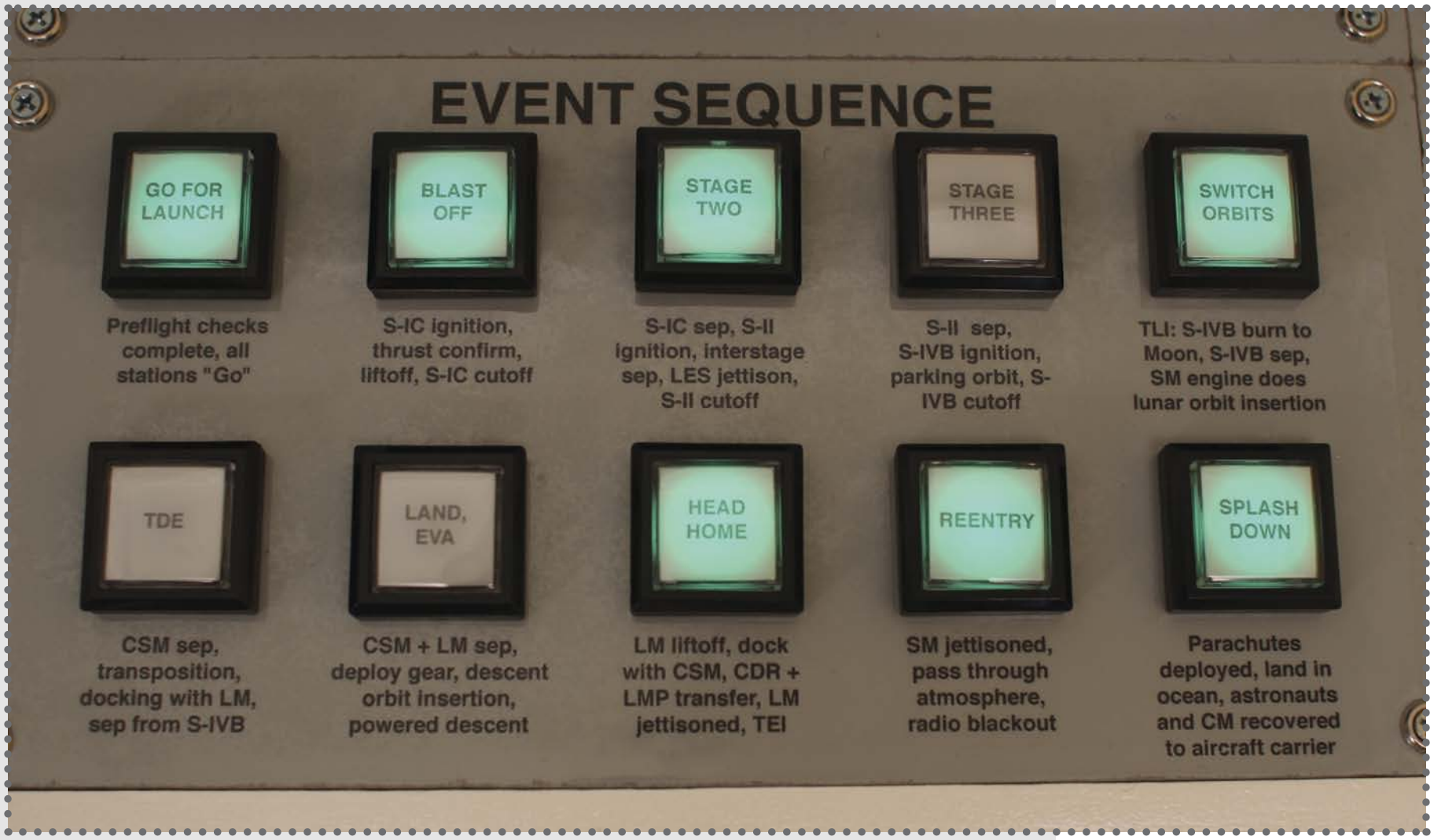
x, y and z in space, and there's one that monitors the astronauts' vital signs (supposedly). There's one that does mechanical spaceship noises, like pumps, heating elements, buzzing noises, fans, etc. I wanted it to be like you're turning things on and off, not just pushing a button that plays a sound – that's why I have the toggle switches as well as push buttons. There's a spot for the iPad in the middle too – you can watch videos of rocket launches.

"There are homages to actual NASA emergencies, like the stirring of the oxygen tanks that led to the Apollo 13 explosion. I have a switch that makes it sound like it's stirring the tanks, then it makes an explosion sound and plays the audio from the astronauts talking: 'Houston – we have a problem'. There's a sequence panel too that has the different mission stages on it and each of those plays a real NASA soundbite, all the way from the launch to the landing on the Moon to the splashdown."

Build steps

Jeff ordered the switches and then designed the panel layout and labels on his computer, printing them on inkjet transparency – clear acetate – and then gluing those onto some fibreboard that he had spray-painted a metallic grey. He estimates that the whole budget was around $700.

Below These reflect the real stages of a NASA mission and play authentic recorded sounds



07

Spaceship Bedroom

After successfully accomplishing his desk mission, Jeff Highsmith set his sights higher

Mission control was but one small step. Next was the mission itself, as Jeff explains: "So my boys would hit the buttons on the desk and go through all those mission stages and run around with their toy rockets, but having the actual spaceship, I thought, would be cool. The spaceship has some panels similar to the desk, but it also has a small screen in there that goes to a video camera in the cargo bay. There's a motorised hatch on the side that you can open up by flicking a switch, and then the camera shows you a cargo bay with a robot arm inside it. You can't see the cargo bay when you're inside because you're laying on your back, but looking at the screen you can see it and the controls are in front of you. It really feels quite fun – I've got a little toy Hubble space telescope in there and I hung a piece of fishing line from the ceiling with a little bit of metal on it, and then I've got a magnet in the space telescope. So you take the telescope from the cargo bay with the arm, move it over and snap it onto the string that hangs from the ceiling – we call that orbit. Once it's in orbit you can pull the arm back in the cargo bay, close the hatch and your mission is complete, you can return to Earth. And then there's the inevitable mission to go and fix the Hubble…

MAKER PROFILE

Jeff Highsmith

Tinkerer extraordinaire

Jeff Highsmith loves to make new and novel things. The medium isn't important, and he enjoys scrounging for materials and making do with what's at hand.

Find out more...

jeffhighsmith.com

Left The astronauts swing their satellites into orbit through the ship's cargo hatch

Jeff's going to upgrade this awesome setup further. "Eventually," he says, "I've got some ceiling satellites planned, so I'll have them orbiting a track in one of the bedrooms and the iPad can monitor the different sensors on the satellites. The track will be a thin metal rod under the ceiling in an ellipse, and then each satellite will have a tiny wheel extending from the top of it, which has a very small gear motor on it, so it'll hang from the track on that wheel. The idea is that the kids can build satellites out of LEGO, put them in the cargo bay, then winch them up into orbit."

"The idea is that the kids can build satellites out of LEGO, put them in the cargo bay, then winch them up into orbit"

08

Automatic Roller Blinds

Nearly a Rube Goldberg machine, we hope it plays 'Powerhouse' when used

There are many schools of thought regarding your sleeping environment to help aid better and more restful sleep. No electronics in the bedroom; try and relax before going to sleep; take a cool or hot shower depending on the time of the year. Some people require pitch darkness to get a good night's sleep, while others like to wake up when the sun rises as part of a natural body reaction. Emil likes to do both of these things with his Raspberry Pi that automatically shuts and opens his blinds at specific times of day:

"7:30: Press up button, wait ten seconds (I have smaller windows on the bottom and bigger above them, so after these ten seconds, shutters are going to be open only on the bottom ones), press stop button. 8:00: Press up button. 22:00: Press down button."

Unfortunately, it means he has some electronics in his bedroom, but whatever works for him.

> "Some people require pitch darkness to get a good night's sleep, while others like to wake up when the sun rises"

MAKER PROFILE

Emil Jaworski
Maker

Finding himself in a rented apartment for a few months, Emil has decided to upgrade it himself.

Find out more...
imgur.com/a/OYdPo

09

Raspberry Pi Terminal

We've had lots of Raspberry Pi over the years, but what about Apple Pi?

"In my spare time I like messing around with older computer hardware that I come across. Originally I found the need for a terminal to connect over SSH to a server that I was using for an internship. It was more efficient than devoting a more powerful machine to it. I took a server to school with me to continue working for the same company as I studied.

I had a Raspberry Pi also sitting around and I started hooking it up to a monitor from an Apple IIc that I found at my university's electronics surplus. This became valuable when I started having to log in to a school server over SSH to compile assignments for my programming classes. Eventually, I found a keyboard from a Macintosh 512k and made it work over USB with a microcontroller and a custom wired key matrix, and I paired it with the monitor and Pi."

MAKER PROFILE

Austen Barker

Engineering student

A Californian student that has a habit of messing around with any old computer hardware that he can get his hands on.

Find out more...

imgur.com/a/vOsML

10

Astrogun

An augmented reality light gun game that turns the real world into your gaming arena

Walking around Maker Faires, you see some weird and wonderful things. At a Jerusalem Maker Faire you may have seen people wield a giant toy gun to shoot down virtual asteroids in Avishay's AR motion game Astrogun. "In the Astrogun lies a Raspberry Pi computer," Avishay explains. "An IMU card connected to it (Sparkfun's MPU-9150 breakout board) gives it the ability to sense the unit's orientation. The Pi is then able to draw the elements seen from that angle. When the player moves, the graphics move, giving the 'object in the room' sense."

Why the Raspberry Pi? It was due to time, according to Avishay: "I had a short time to bring it to a working thing, so I had to pick a platform that was capable of the task and easy to use. The RPi fits that criteria. I used many software components designed for the RPi or tested on it – the Pi3D and RTIMULib. The combination of Pi as a hardware platform and Python as a programming language is the fastest way to materialise ideas."

MAKER PROFILE

Avishay Orpaz
Electronics engineer

Find out more...
bit.ly/1AYPSqg

Above
Time Crisis ain't got nothing on this beast of a light gun

11

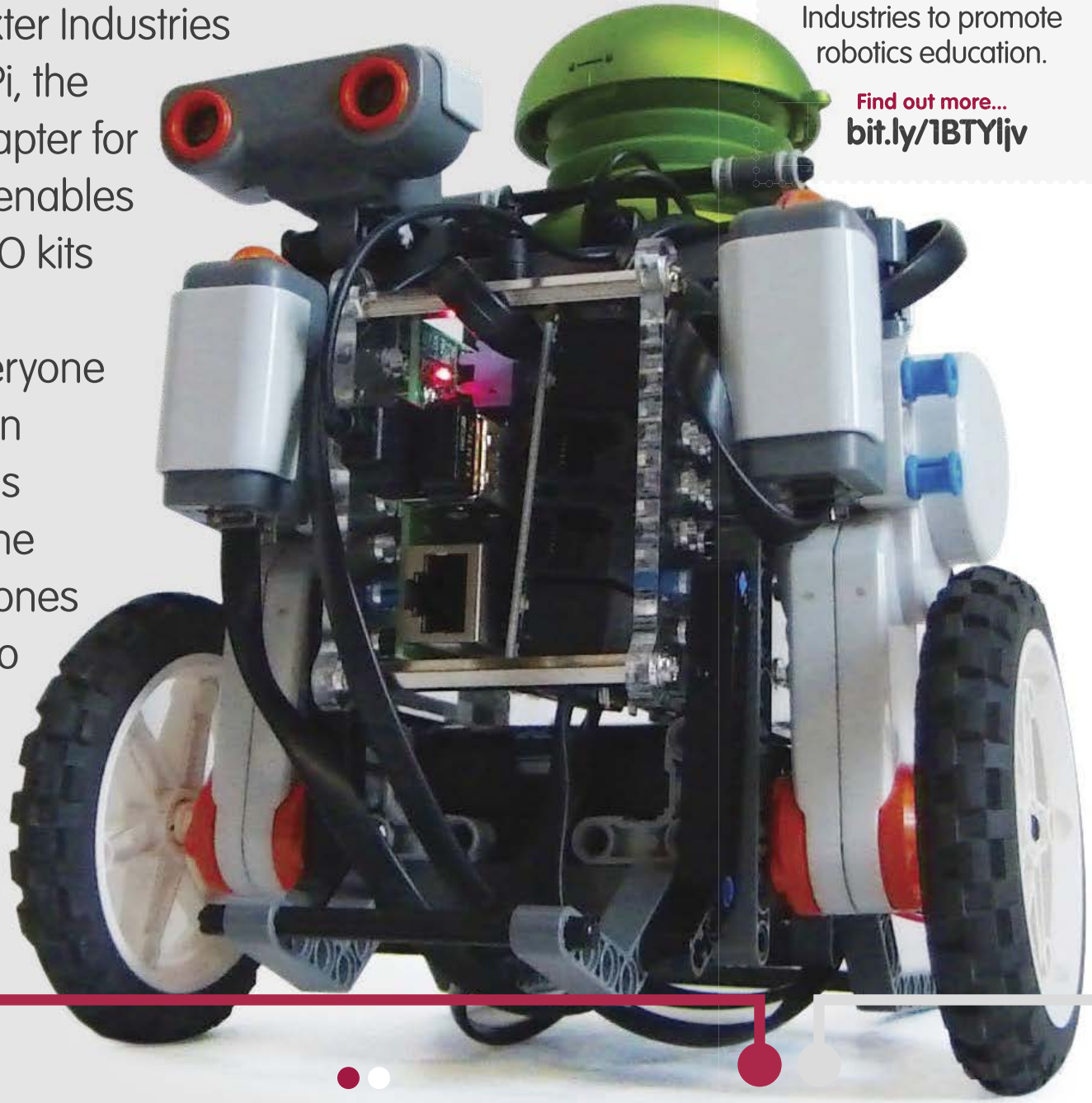
Alarm Clock Robot

Chasing your alarm clock may sound like a nightmare to some, but here it is in reality

When we talked about Rolly the alarm clock robot around the office, most people burst forth with a string of expletives not fit for print. It's a delightfully evil invention – an alarm clock you need to work for to turn off. It sounds like a great invention, getting people who have trouble waking up to actually get up out of bed and start the morning.

Like any other Dexter Industries robot, it runs on BrickPi, the LEGO Mindstorms adapter for the Raspberry Pi that enables it to interface with LEGO kits via programming.

"Today almost everyone uses their phone as an alarm clock, which has a range of benefits," the website explains. "Phones are easy to set, easy to



Right Because getting out of bed in the morning isn't half as fun as it could be

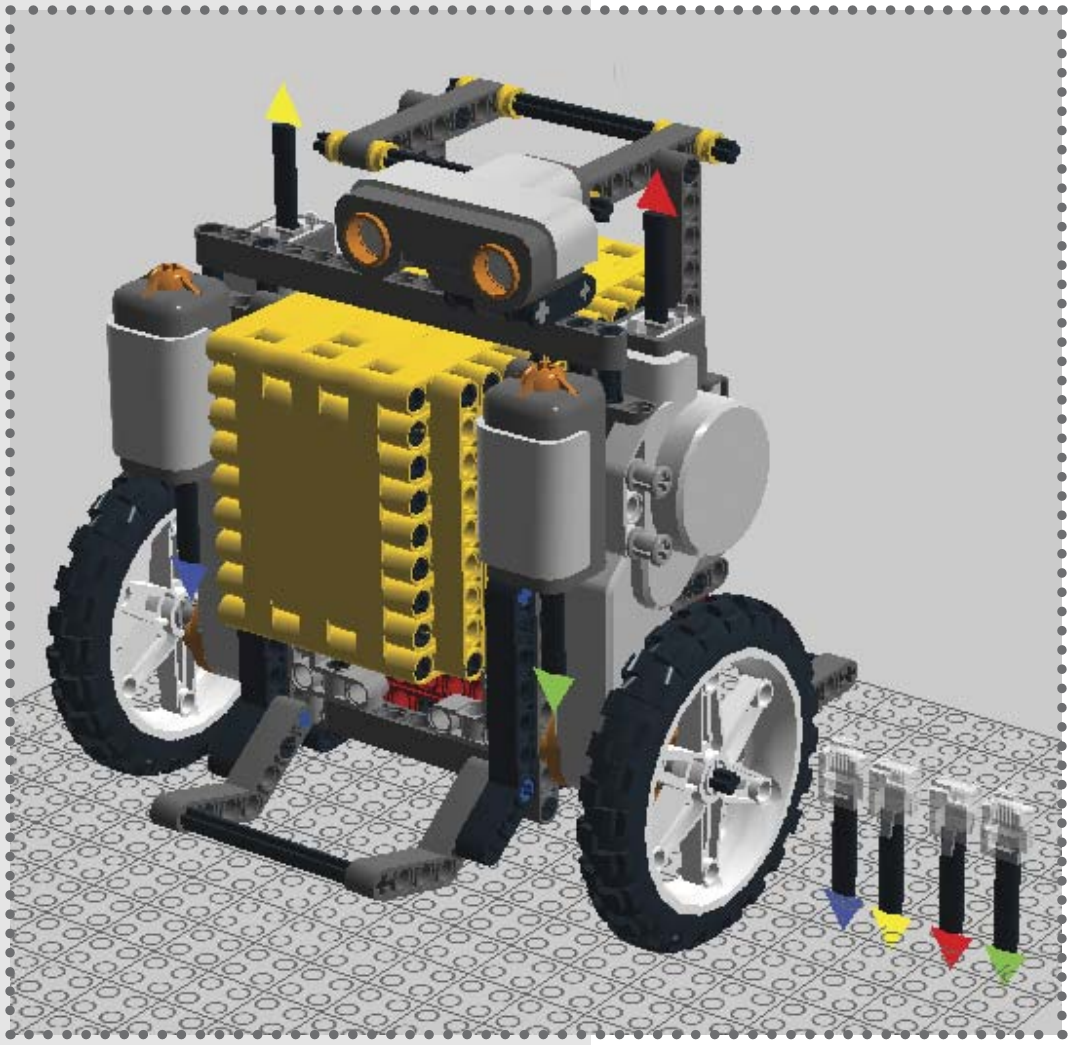
MAKER PROFILE

Taryn Sullivan
Advisor

Taryn is an international businesswoman. As well as flying between Shanghai and DC for her own engineering business, she now works with Dexter Industries to promote robotics education.

Find out more...
bit.ly/1BTYljv

> “Simply link the program on the BrickPi to your Google account and it will search events with the title ‘wake1’”



adjust, play custom songs and can even sense when is the best time to wake you up. The problem is, unless your phone is across the room, we use our phones so much we can literally use them in our sleep. Why not build a robot that is able to do all these things, but won’t stop till you get up and start moving! Our robot will be able to easily move randomly around the room over any surface, playing a custom alarm tone.

“In order to set the alarm, simply link the program on the BrickPi to your Google account and it will search events with the title ‘wake1’ and automatically start the alarm at the event’s time. This means that the alarm time can very easily be adjusted using any device that can gain access to your Google Calender.”

Robotics education

There are many ways to try and get kids excited with coding, like making games in Scratch, modifying *Minecraft* and teaching via robotics. The latter is a newer concept but still has the same merits – a physical creation that children are excited about then react to the programming they’ve done on it. Visible results and instant gratification is a great way to get imaginations fired up.

Left The Alarm Clock Robot is built to be sturdy, which it will need to be to withstand undercaffeinated alarm stoppers

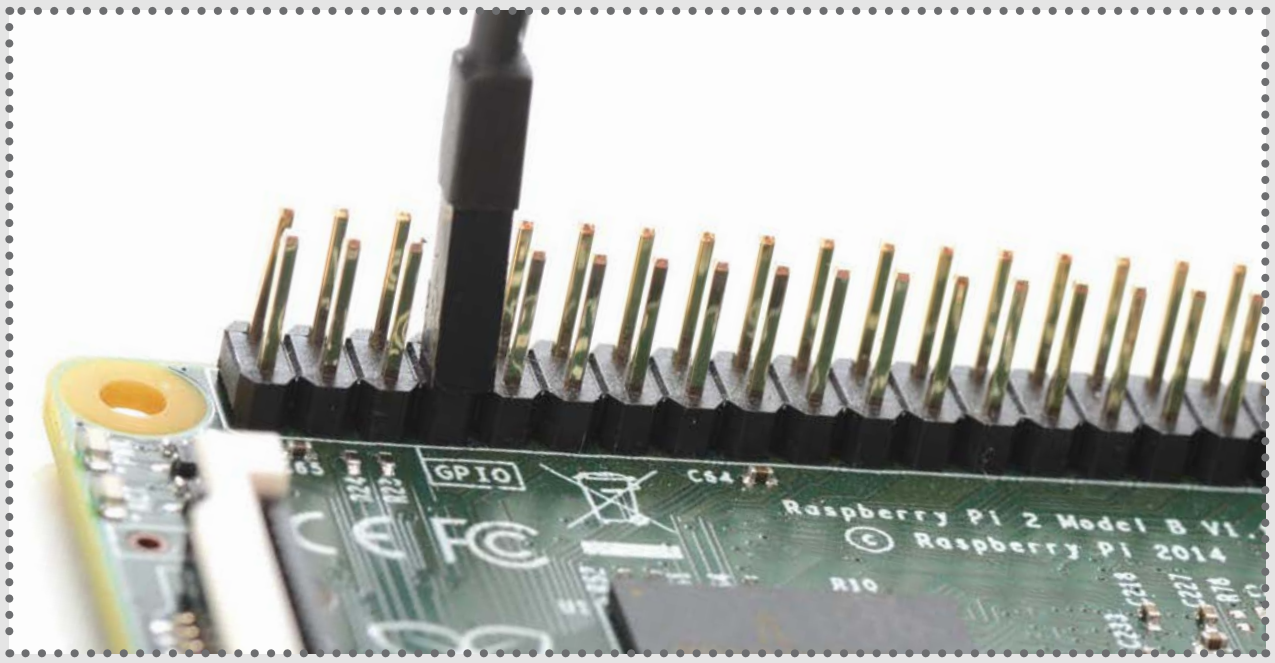
12

PiFM Radio

Sometimes the simplest hacks can open up whole worlds of possibility

When studying at Imperial College London, Oskar Weigl and Oliver Mattos discovered how to the Pi into an FM radio by connecting a wire (antenna) to GPIO 4 and using a custom Python module.

"There is a clock generation module in the hardware of the Raspberry Pi chip that lets you output a clock signal at a user-selected frequency," Oskar explains. "We used the DMA controller to send commands to the clock module to change the frequency and achieve frequency modulation. We had to overclock the clock generation module by a factor of 20 times. The sound is 14 bits per sample, enhanced to a higher number of bits using delta sigma modulation and the range is at least 50 metres."



MAKER PROFILE

Oskar Weigl
Electronics engineer

Oskar is an electronics professional and hobbyist, as well as an avid forward and reverse engineer.

Find out more...
bit.ly/1om6BQE

Left
Want to make your own? Check out issue #156 of **Linux User & Developer** magazine!

13

Ras Pi Smart TV

Is your smart TV not smart enough? Open the case and put a Pi inside

"There's plenty of room for additional electronics inside the Hisense LED smart TV," begins Tony Hoang. "There's a large flat area for electronic parts in the centre of the TV where I placed my Raspberry Pi. The dual down-facing speakers were quite loud, so I removed one and replaced it with a USB hub. The back panel was mostly flat, so finding a spot for the LAN port and HDMI output wasn't too hard.

"The Raspberry Pi is powered by the logic board of the Hisense. There were the obvious 5v-500 mAh outputs from the 2x USB 2.0 ports, which I tried but I found out that the logic board shuts off the power to these ports when the TV turns off. To keep the Raspberry Pi turned on, I probed the logic board with a multimeter and found one from an unused GPIO."

MAKER PROFILE

Tony Hoang
Graduate researcher

Tony Hoang is a PhD student studying computational biophysics and single molecule research at SUNY-Albany in Albany.

Find out more...
linkedin.com/in/tonyphoang

14

Pirate TV

Go all the way and totally rebuild the Android TV platform for a totally custom setup

The perfect companion to the Raspberry Pi Smart TV hack (on the previous page), Donald Derek Haddad's project is a custom TV interface that you can make yourself.

"Pirate TV is a smart television application that runs on the Raspberry Pi with the Raspbian OS," Donald tells us. "It's built with open source tools and shipped with a free remote controller, your mobile device. At its core lies a Node.js application that runs a web server with Express.js/Socket.io to handle users requests from the remote and trigger shell scripts. The TV user interface is rendered on a Chromium instance in kiosk mode. Videos are streamed from YouTube or other channels played on OMXPlayer, and cached including 1080P HD content. This project is a work in progress and it's not going to be able to tap into a lot of the content, which makes a Google (now Android) TV or other commercial platforms so valuable." Check out Derek's tutorial: **http://bit.ly/1l6YKpj**.

MAKER PROFILE

Donald Derek Haddad
Software engineer

Donald is an open source hacker.

Find out more...
donaldderek.com

Above
Fully open source, this is Donald's Pirate TV interface

15

Pye Radio

Old meets new in this modified radio, upgraded with Wi-Fi to play internet radio stations

Upcycling is a great concept: recycling a product using new technology to make it relevant in the modern world. While standard analogue radio isn't dead yet, its nice to have options when listening to music. This is where Tony's idea came in:

"I'd been working on a Raspberry Pi to play music streams through my stereo. Once this was running I integrated the Raspberry Pi into an old radio named a Pye! With a background in woodwork and engineering this seemed like the perfect project."

However, this project wasn't a walk in the park for Tony: "The hardest part of the conversion is linking the tuner knob to the rotary encoder. For this radio I used the spring from an old biro as a drive train to link the tuner knob spindle to the rotary encoder through 90 degrees."

Did Tony have qualms about heavily modifying such a classic design? "None whatsoever! Some people don't like the modern style now stamped on this old radio, I think it's a new era, new look!"

MAKER PROFILE

Tony Flynn
Senior embedded systems design engineer

Find out more...

bit.ly/19zrgPI

Below
The radio uses text-to-speech to read out station names

16

Digital Camera Conversion

Classic aesthetics with modern convenience, this old-school camera has been upgraded with a Raspberry Pi

Old cameras have a very specific design aesthetic that it seems has been lost to time, although nostalgia for them is still very strong in certain circles. So if you like the aesthetic and aren't too bothered about using the old photo process, why not upgrade the insides with more modern technology?

"I wanted a suitable case for the Raspberry Pi camera board," Pete tells us, "and the Holga seemed a perfect fit. Most of the available cases are either a bit ugly or suited

Left The camera of the past, updated for today

MAKER PROFILE

Pete Taylor
Web manager

Pete works for a charity and has always tinkered with computers, ever since he got a hand-me-down BBC computer. He likes that the Raspberry Pi returns to a time when you could hack your own computer without making an expensive mistake.

Find out more...
bit.ly/1MKRASw

Left The Pi is the perfect size to fit inside the original camera case

more towards stationary webcam-type applications." Why the Raspberry Pi, though? "The Raspberry Pi is a maker's dream – it's cheap and cheerful, and the community that's built up around the Pi makes a brilliant resource when you're stuck with a problem or want to find out more."

As well as the actual camera and Pi itself, Pete used a Raspberry Pi camera board to actually take photos, a Wi-Fi module for connecting to it remotely, a battery and a switch for it, and a few buttons and resistors to wire the camera's control buttons to the Pi.

"It works better than I expected!" Pete said about the quality of the finished product. "Although it seems a bit daft to build a camera that's about as good as you'd get from a cheap camera phone, it's changed the way I take pictures. By removing the instant replay – most people seem to view the world through the displays on their phones – I can concentrate on taking the photo. Only seeing the pictures when I've taken the camera home and downloaded them to my PC adds a bit to the film nostalgia and I'm often surprised by the photos I've taken. Plus I've received some nice feedback about how the camera looks."

Do it yourself

Version two of the project will result in a kit that people can use to convert their own cameras "that doesn't require you to take a Dremel to the insides of a Holga!". He's not decided yet on whether to make a kit that converts a Holga, or a kit that builds a Holga-esque case around the Raspberry Pi itself. Either way, the whole thing should also have a better photo-taking capability, which is the ultimate goal.

17

Bitcoin Pool Table

Modern pool pros pay digitally by scanning a QR code to insert Bitcoins and rack up a game

Liberty Games loaded a Raspberry Pi into the side of a pool table that enables people to make payments via a Bitcoin app on their phone to release the balls – very clever stuff.

"We did this with the Raspberry Pi 1", Stuart tells us. "We tried to install the entire Bitcoin client on the Pi but it was struggling, and downloading the whole blockchain caused issues too. It was going to be connected to the Internet, so we offloaded the heavier work to a server able to handle the blockchain. The server receives the Bitcoin payment and then communicates to the Pi securely that the payment has come in, and it syncs up the prices as well.

"We connected a PiFace via a breakout board that's monitoring on WebRCT for the go-ahead from the server, and once it gets that, it sends the physical voltage to the electrical ball release mechanism."

Above The price of a game is displayed, in Bitcoins, on the PiFace display module

MAKER PROFILE

Stuart Kerr
Technical director

Stuart Kerr is the technical director of Liberty Games, which specialises in classic table and bar games. You might have heard of some of their hacks.

Find out more...
libertygames.co.uk

Fireball Pinball

Ian Cole and his sons enter the high score boards with Fireball HD

At **http://raisinggeeks.com**, Ian Cole and his two sons love a challenge. So the chance to repair a game table gave them the perfect Pi primer.

"We've taken an existing pinball machine playfield," says Ian, "and built a new game from it. This required learning the underlying hardware first. Then we learned how to use the Raspberry Pi, pygame for sound and text graphics, and omxplayer for HD video, and we connected the software tools with the hardware of the pinball machine.

"We built a MOSFET circuit on a breadboard to test a single solenoid. When that worked, we duplicated it onto a hand-soldered protoboard and extended it to control the five solenoids. The Raspberry Pi handles graphics, audio, scoring rules, saving scores, etc. One Arduino drives the lamp and switch matrix, another drives the solenoids. The three are connected with an I2C bus."

MAKER PROFILE

Ian Cole
Maker and geek raiser

Ian Cole is a keen maker, hacker and inventor, and regularly blogs about his family projects with his two sons. Their Fireball project, for example, grew out of an innocent 'Can you make it playable?'

Find out more...
raisinggeeks.com

19

Voice-Controlled Lighting

Light up your living room with a voice-controlled light show for a coffee table

Mikel found himself in a quandary: he didn't have a coffee table. However, instead of just heading to Ikea, he decided to make one himself – one that lit up using a Raspberry Pi to control the sequence.

"I used the Raspberry Pi because it was pretty easy to work with and has a great community. Whenever I run into an issue, there is always some documentation on how to fix it. I wanted to have network support and do more complex operations, like loading images, than I could do just using an Arduino with Ethernet. The combination of the two really made development faster. Now that Pi4J has added support for the Raspberry Pi's integrated SPI, I am working on controlling the LEDs directly, without the Arduino."

Recently, Mikel added voice control to it using Google Now. "After you say a command to Google Now, Autovoice, a plugin for the Android app Tasker, intercepts the response back from Google. If the response matches a Tasker task, the task will be launched. I have a few tasks set up to make web service calls, basically calling a URL with parameters. This URL connects to a Java based web service I created, running on my Raspberry Pi." Mikel encourages users not to use coasters, as drinks light up while the table is switched on.

MAKER PROFILE

Mikel Duke
Software developer

Mikel loves programming, photography, biking and hiking, and sharing his projects.

Find out more...
bit.ly/1F0HDkW

Project Jarvis

The 1950s future today with your voice-activated smart home automation system

Home automation has been a thing for years now, with a fair few intrepid engineers and DIYers modifying their home. With wirelessly controlled lights, heating, garage doors and many other household items, it's amazing what some have achieved. Science fiction has always portrayed houses with advanced systems and Mayur has been busy creating such a system.

"The major inspiration came from watching the *Iron Man* movies, which is where the name Jarvis comes from. I decided to build a simple home automation system, however over the many years and many reinstalls of

MAKER PROFILE

Mayur Singh
Computer systems engineering student

Located in South Africa, Mayur's expertise is in embedded systems and he is familiar with microcontrollers and systems that enable outside hardware interaction, such as the Raspberry Pi and the Beaglebone Black.

Find out more...
bit.ly/1Evlmci

Left The Jarvis interface is gorgeous, but you can control the system from anywhere using your smartphone or simply your voice

Windows, I lost my program. I built a new and improved home automation system, which features an AI assistant and more functionality, after I saw the third *Iron Man* film."

While home automation is one part of the project, convenience isn't the only factor.

"The main function of the system is to help save energy in homes," Mayur tells us. "Jarvis can read and monitor the electricity usage per light or appliance and this lets the AI perform certain tasks. These tasks are determined by outside factors, like whether or not a light should be switched on if there is adequate natural light in the room. This is a basic example but other factors influence the determination of the AI, which has control over the power to each light or appliance. The data is logged throughout the month and the system uses that information to achieve better results the following month. This is smart home automation."

The Jarvis AI is a major part of it, and you can talk to it. "He can control your home using voice interaction and make basic decisions about energy savings in a room. I have also recently built a wall-mounted tablet system that links to a local online grocery store. It uses voice control to identify what products you need and adds the items to a list until you specify the delivery."

With a year and a half of work on the project, everything Mayur has completed works reliably, but there are still more functions he wants to add.

Above Alarm mode can only be disabled with the right finger or the master code

The voice

The Jarvis that this project is named after is well-known for its snarky remarks and tone of voice. Choosing an actual voice that works though is a hard enough task without going for a certain style, as Mayur explains:

"It's been difficult trying to find voices with a good API and price but I have settled for a free API which offers cloud conversion and just sends an MP3 sound file back. It takes longer but it ensures that the voice works on all operating systems."

Tether your Raspberry Pi to an Android device

Need the internet on your Pi while on the go? Try out a physical tether to your Android device for instant online access

The portability of the Raspberry Pi is one of its most lauded features and you can get many different accessories to help aid this portability. Mini screens, mini wireless keyboard and mouse combos, portable batteries, cool cases and more can get you out and about, but the internet is a stumbling block that you can't easily fix with an accessory. However, what you do also usually have with you is an internet-connected magic pocket box called a smartphone that, with a bit of know-how, you can connect your Raspberry Pi to and steal some internet from. Over the next few pages we will impart this know-how to get you using your Raspberry Pi on the internet when you're on the go.

THE PROJECT ESSENTIALS

Android device

USB cable

01 The easy way

A lot of smartphones now have a Wi-Fi hotspot feature, which the Raspberry Pi can easily attach to. First of all, turn the hotspot on and then boot into the Pi. Connect a wireless dongle and open up the wpa_gui in Preferences>Wi-Fi Configuration.

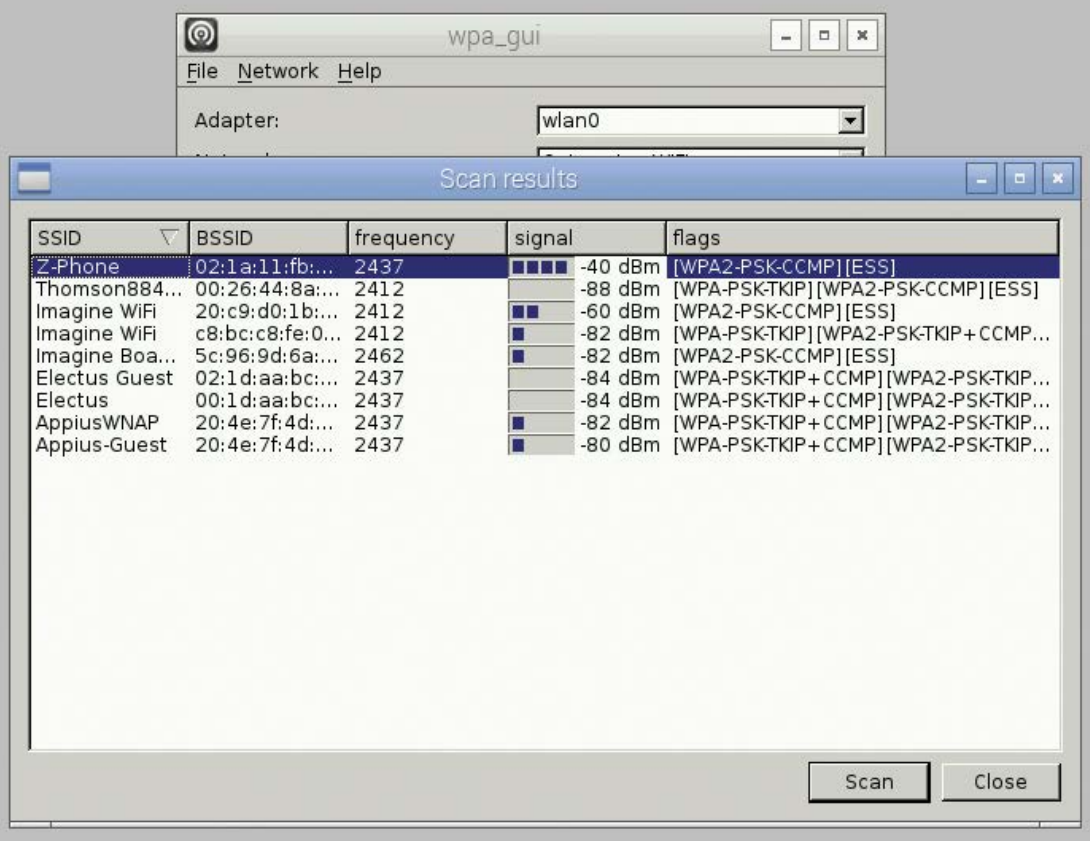
02 Scan for device

Click Scan to open up the scan window and then select Scan again from inside there. It should pick up your device – connect it as you would to any Wi-Fi network and the Pi will remember it for when it needs it next.

03 Set up tether

First connect your phone to your Raspberry Pi via a USB cable – depending on the amount of power your

"Turn the hotspot on and then boot into the Pi. Connect a wireless dongle and open up the wpa_gui"



Left It's usually easy to figure out which device is your phone – set it next to the Pi and it will be the one with the best signal!

Pi has, it might have trouble charging your phone but it will still let you tether. In the tethering menu you can now activate USB tethering.

04 Check connection

Your Android device will create an interface known as eth0 on the Raspberry Pi. You can check to make sure this is happening, and that it will let you tether, by opening up a terminal and typing the following:

```
$ ifconfig
```

05 Quick connect

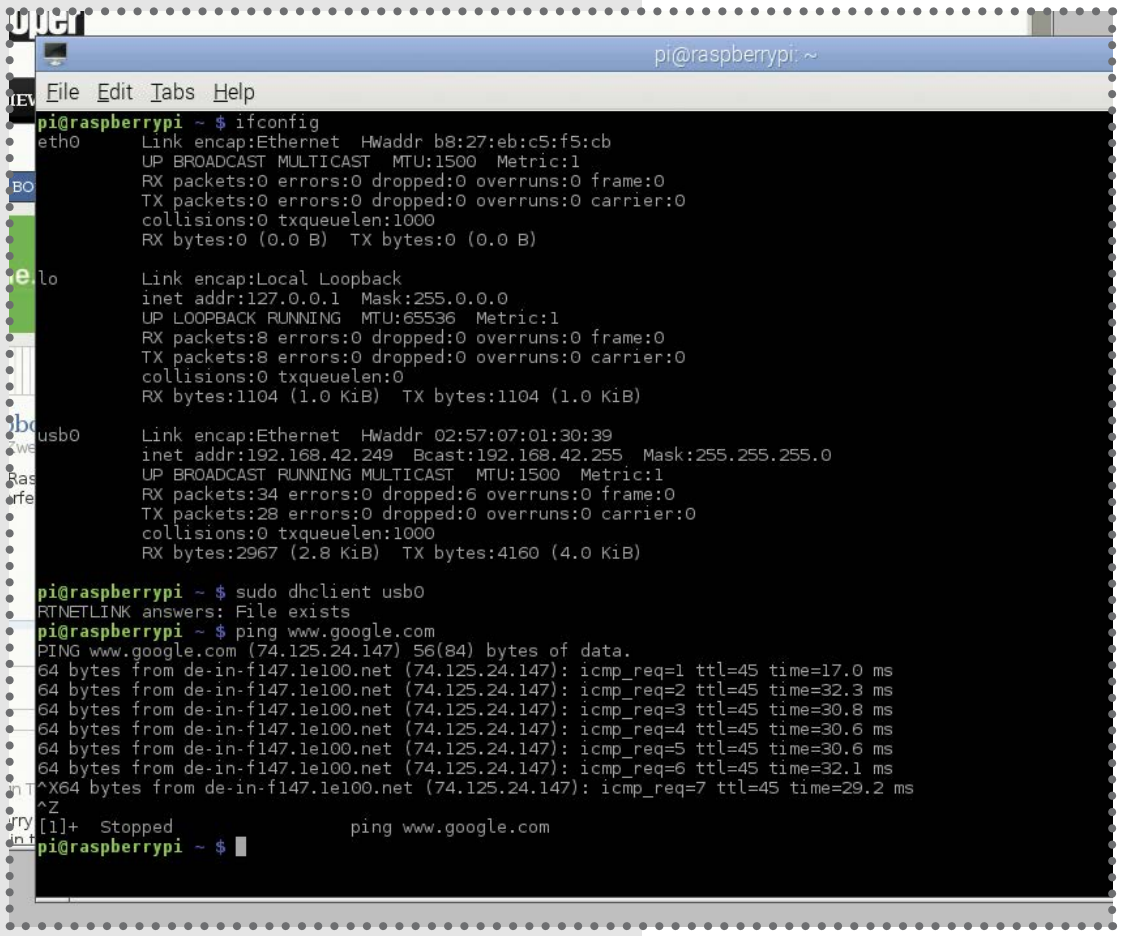
You can connect from the terminal right now to access the internet. You should be able to do this by typing the following into the terminal:

```
$ sudo dhclient usb0
```

This will automatically grab any available IP address that your phone will give to it.

06 Test connection

There are a few ways to test your connection. We'd usually stay in the terminal and ping Google, which you can do, or you can click on the browser and see if it loads the page.



Above Assuming that dhclient reports 'File exists', you should be good to go

07 Save the settings

Once you reboot your Pi, it won't remember to automatically connect to the phone's tether. However, we can add an entry to its config so that it will try and do this in the future. From the terminal use:

```
$ sudo nano /etc/network/interfaces
```

08 Interface settings

Here you'll find all the current network settings – yours might look different from ours depending on if you have added any fixed wireless settings or passthroughs. Using the same syntax as the eth0 line, add:

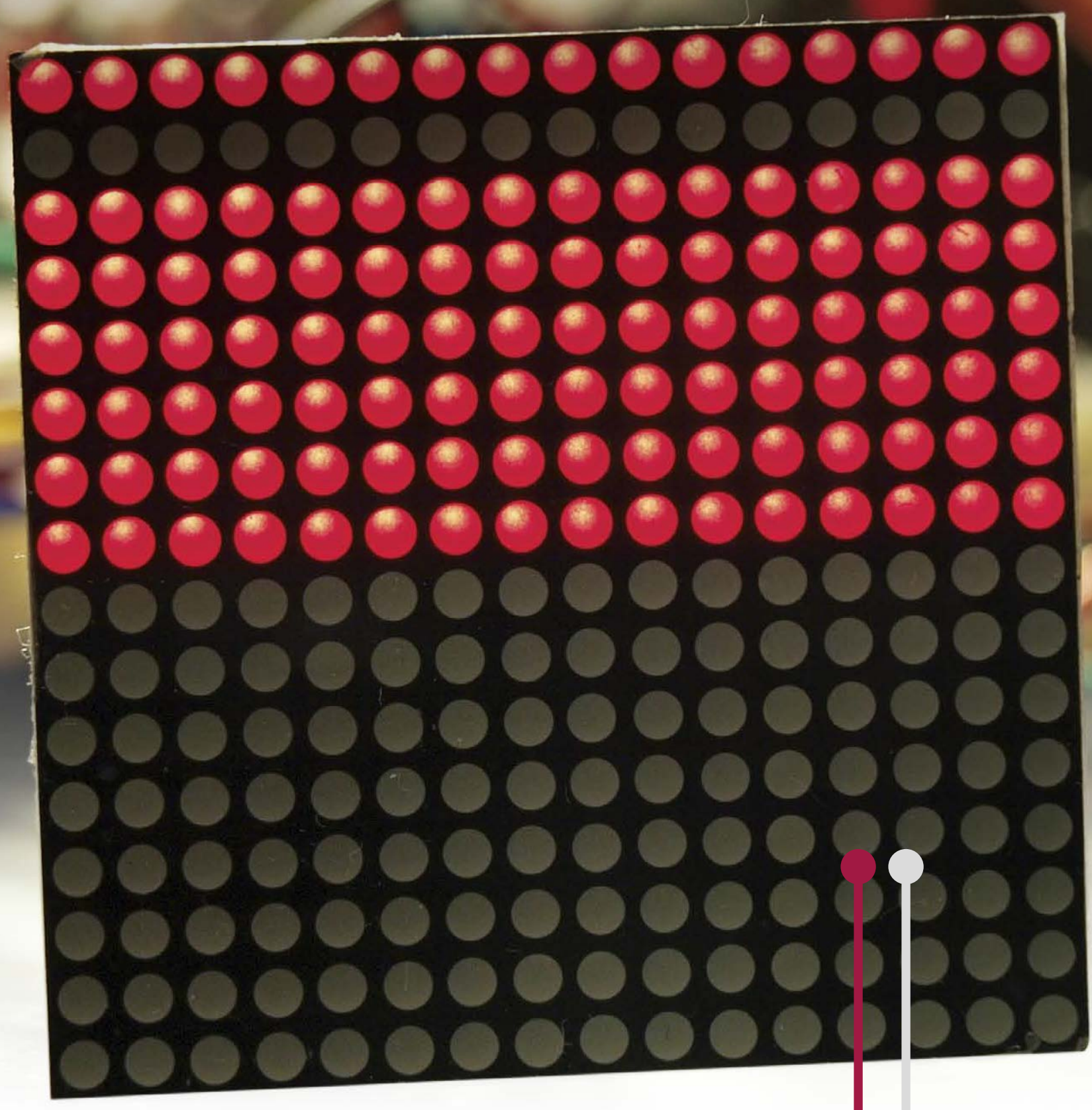
```
iface usb0 inet dhcp
```

09 Tether on the go

After a save and reboot, your Pi should now automatically connect to your phone, whether it's via Wi-Fi hotspot or a physical connection. It may draw a little more charge than usual while tethering, so be sure to keep an eye on your battery level.

Build a complex LED matrix

LED Matrix display systems find use everywhere from gaudy kebab shops to impressive steampunk systems



Driving LEDs in an efficient fashion is a science of its own. The common availability of single-board computers has put the necessary technology within reach of everyone.

When dealing with LED displays, two different systems must be considered. We will focus on traditional matrix-based systems made up of one or more LEDs. Their affordable nature makes them ideally suited to classic display applications: they communicate currency prices, provide stock-brokers with updates from the trading floor and have even been used as basic displays for primitive oscilloscopes.

Finally, we will also provide you with an overview of electronic basics. This tutorial is a bit more advanced than the ones we usually run in this section of the magazine, and it's also worth noting that we're going to be programming with C rather than Python. Follow along using the code listing annos.

THE PROJECT ESSENTIALS

Breadboard & wires

16x16 LED matrix

2x 74HC238

2x 74HC244

16x 220 Ohm resistor

Source code

http://bit.ly/1Tcpo1H

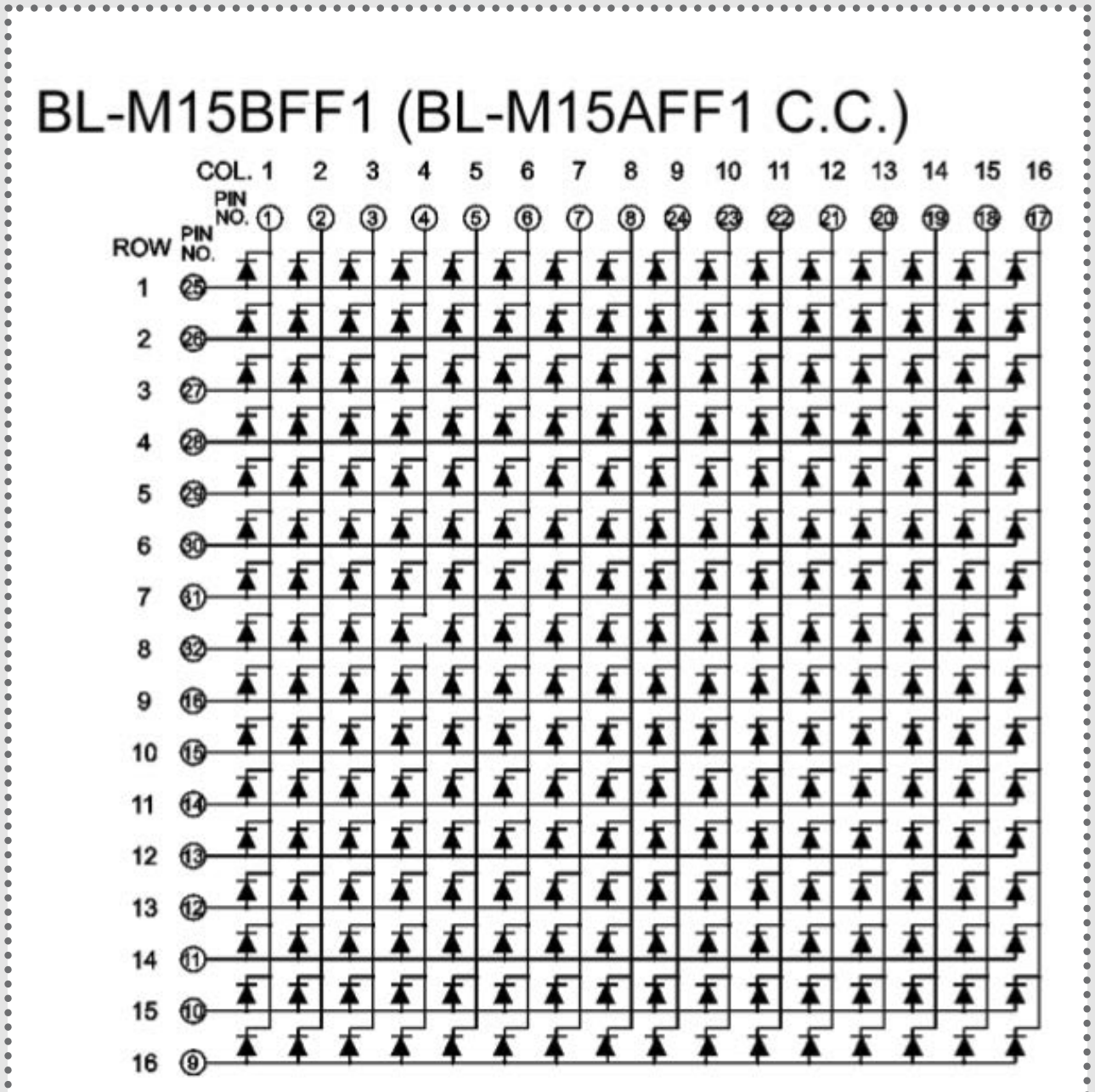
01 Think about LEDs

Standalone LEDs are primitive – they light up once current flows through them. Driving a few LEDs is as easy as connecting them to GPIO pins along with a resistor. Sadly, this method becomes wasteful once more than a few of them get involved – driving 16 diodes ties up 16 pins.

02 Arrange your diodes

Methods were devised to reduce the number of pins needed. Matrix-based systems are resilient to individual diode failures, and provide a pin-to-LED ratio of n=(n/2)^2. The following steps assume a

"Driving a few LEDs is as easy as connecting them to GPIO pins along with a resistor. Sadly, this becomes wasteful once more than a few of them get involved"



16x16 LED matrix which is made up according to the schematic above. Since LEDs permit current in only one direction, you can enable a single LED by bringing the corresponding pins high and low.

03 Harness the MUX

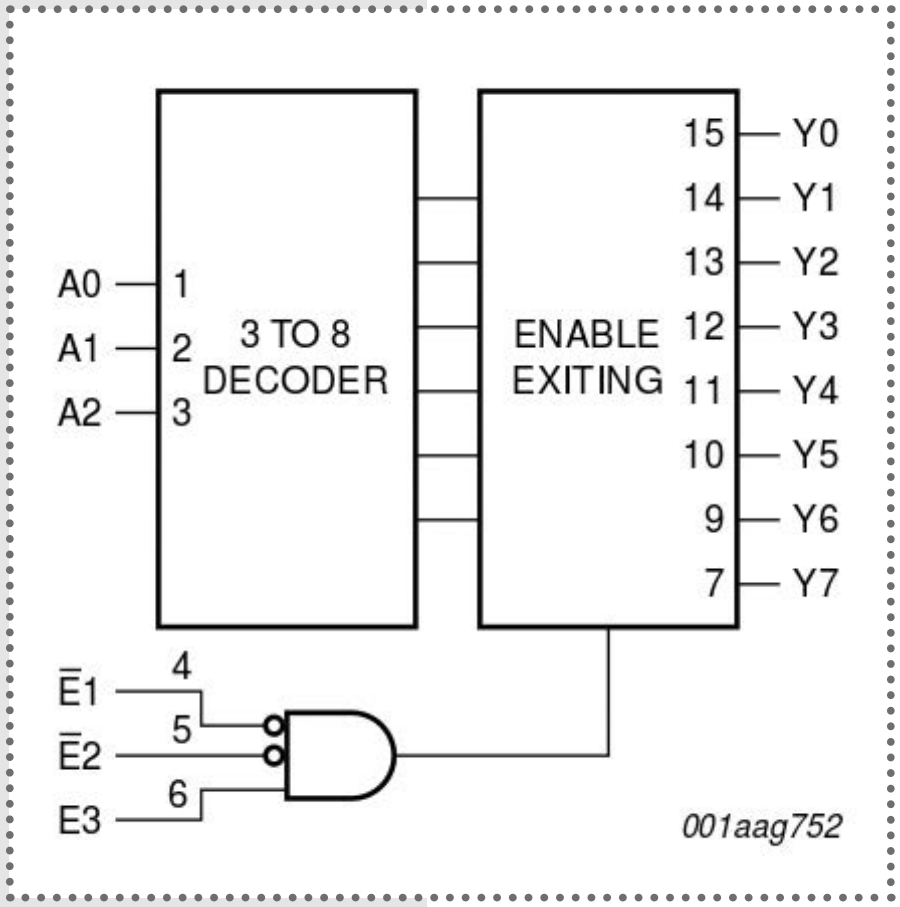
Our LED module has a total of 32 inputs, which overwhelms older versions of the RPi. The first way to restrict their number comes in the shape of the 74HC238, a component described as a 3-to-8 line decoder/demultiplexer. Its function is described in the schematic on the next page.

Left The extended version of this schematic is inside your source code pack at – get it at **http://bit.ly/1Tcpo1H**

"Our LED module has a total of 32 inputs, which overwhelms older versions of the RPi"

04 Separate concerns

Chip two goes by the name of 74HC244, which is described as an octal buffer with tri-state capability. Tri-State outputs can physically disconnect themselves from the bus line. This permits you to tie their outputs together without fear of short circuits. As long as all but one chip are in tri-state mode, no current can flow between high and low output pins.



05 Round them up

Four GPIO pins control the currently-enabled 'line' of the display. Three pins configure the address which is to be emitted, while the signal emitted from the fourth pin is connected to the activity inputs. This ensures that but one IC is active. The 74HC244 ensures that but one of the two groups is active at any given time.

Above Three binary outputs determine which of the eight outputs will be set to High, thus expanding your total outputs

06 Configure the pins

We used a library from Hussam Al-Hertani's blog (**http://hertaville.com/2014/07/07/rpimmapgpio**). The first step involves setting output functions. As the GPIOs are set to outputs, the tri-state feature might connect the internal state to the output pins of the IC. This could lead to internal shorting if the output is not turned off.

07 Power the MUX

Create a convenience function taking an address ranging from zero to 15. It is converted into pin outputs for our 3-to-8-demultiplexer. The effect of this is that all but one of the sixteen rows is to be supplied with energy.

08 Select a row

In the 74HC244, we first disable both units and proceed to turning on the one which is needed. This sequence prevents ghosting during the switching process.

09 Do the main loop

The outer part of the loop consists of logic that manages the addressing of the individual rows. Our program must flash the individual LED groups one after another by using the building blocks described in the next step.

10 Complete the loop

Writing out data is accomplished in a sequence of three commands. We select the row, configure the column and then write out the data bits that are to be displayed. A small pause is observed in order to give the LEDs some time to 'burn into' the viewer's eyes.

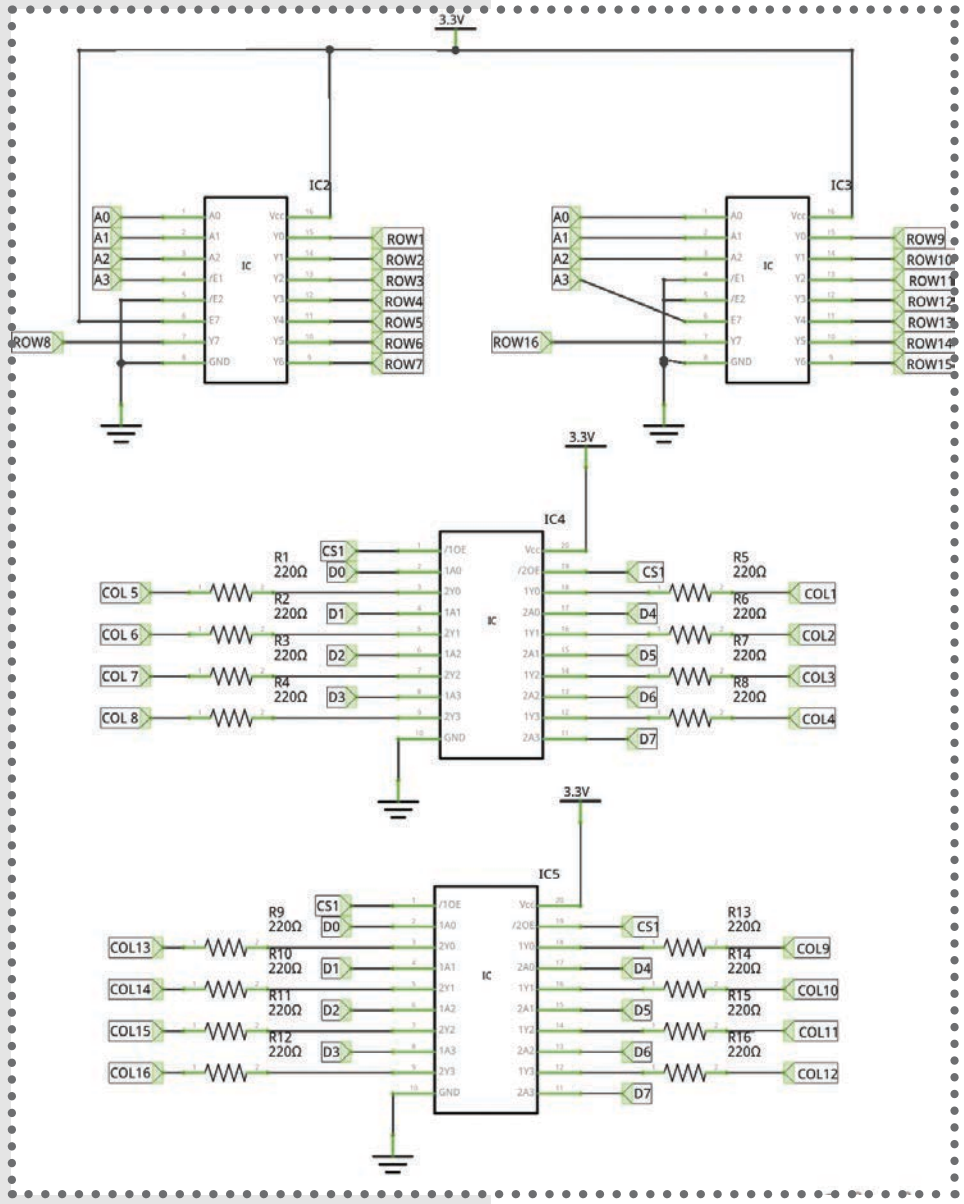
11 Energy control

LEDs light up if current flows through them. SetData pulls the pins of the 74HC244 low to ensure that the energy supplied from the 74HC238 flows through the diode.

12 Avoid GPIO trouble

The Raspberry Pi Foundation has been known to change the layout of the expansion header, a thing which professional manufacturers

Below This is the full schematic of the LED matrix that we're working with here (full-size version also available in your pack)



of process computers abhor. It's recommended to handle the mapping between pins and functions via a set of defines. Our code is optimised for a Rev2 Raspberry Pi with a 'short' header – 40-pin variants will require readjustments making sure the physical pin numbers correspond to the logical GPIO numbers.

13 Add example data

Test the code by setting the datastore to a value of your choice. Setting 64 to all fields will disable one row in each part of the display.

14 Kick it off

Check all connections between the planar and the single-board computer, and proceed to starting the compiled app. Don't forget to use the **sudo** command – direct memory access is restricted to root in order to prevent apps from causing havoc in the physical memory. Users are accustomed to this, so requiring them to put a sudo in front of the command doesn't cause concern.

15 Notice a flicker

Sharp-eyed readers will notice an occasional flicker in their LED matrix where one line appears brighter than the others. This is caused by the stalling of the program – if the kernel does other work, the switching routine can't run. We could solve this problem by using a real-time Linux kernel.

"Our code is optimised for a Rev2 Raspberry Pi with a 'short' header – 40-pin variants will require readjustments"

The Code LED MATRIX

12

```
#include “mmapGpio.h”
#include <unistd.h>
#include <stdio.h>

#define PINA0   2 // 3
#define PINA1   3 // 5
#define PINA2   4 // 7
#define PINA3   14// 8
#define PINCS0  17// 11
#define PINCS1  18// 12

#define PIND0   23// 16
#define PIND1   24// 18
#define PIND2   10// 19
#define PIND3   9 // 21
#define PIND4   25// 22
#define PIND5   11// 23
#define PIND6   8 // 24
#define PIND7   7 // 26
```

07

```
void setAddress(unsigned char _which, mmapGpio* _where)
{
  if(_which&1)
  {
    _where->writePinHigh(PINA0);
  }
  else
  {
    _where->writePinLow(PINA0);
  }
  if(_which&2)
  {
      _where->writePinHigh(PINA1);
```

07

```
  }
  else
  {
      _where->writePinLow(PINA1);
  }

  if(_which&4)
  {
       _where->writePinHigh(PINA2);
  }
  else
  {
      _where->writePinLow(PINA2);
  }

  if(_which&8)
  {
      _where->writePinHigh(PINA3);
  }
  else
  {
      _where->writePinLow(PINA3);
  }
}
```

08

```
void safelySetRow(unsigned char _which, mmapGpio* _where)
{
  _where->writePinHigh(PINCS0);
    _where->writePinHigh(PINCS1);
  if(_which==0)
  {
    _where->writePinLow(PINCS0);
  }
```

The Code LED MATRIX

08

```
  else
  {
      _where->writePinLow(PINCS1);
  }
}
```

11

```
void setData(unsigned char _which, mmapGpio* _where)
{
    if(_which&1)
    {
        _where->writePinHigh(PIND0);
    }
    else
    {
        _where->writePinLow(PIND0);
    }
    if(_which&2)
    {
        _where->writePinHigh(PIND1);
    }
    else
    {
        _where->writePinLow(PIND1);
    }

    if(_which&4)
    {
        _where->writePinHigh(PIND2);
    }
    else
    {
        _where->writePinLow(PIND2);
    }
```

```
    if(_which&8)
    {
        _where->writePinHigh(PIND3);
    }
    else
    {
        _where->writePinLow(PIND3);
    }
    if(_which&16)
    {
        _where->writePinHigh(PIND4);
    }
    else
    {
        _where->writePinLow(PIND4);
    }
    if(_which&32)
    {
        _where->writePinHigh(PIND5);
    }
    else
    {
        _where->writePinLow(PIND5);
    }
    if(_which&64)
    {
        _where->writePinHigh(PIND6);
    }
    else
    {
        _where->writePinLow(PIND6);
    }
    if(_which&128)
```

LED stripes

Two versions of LED strips are offered. 'Primitive' ones are based on analogue technology. In it, an entire strip of diodes has the colour set by the three input pins. Systems such as the mega-display shown in the left-hand image require the use of the digital version. They are based on the concept of the shift register. Your system inputs individual colour values which are then pushed on along the strip.

The Code LED MATRIX

```
    {
        _where->writePinHigh(PIND7);
    }
    else
    {
        _where->writePinLow(PIND7);
    }

}
```

06

```
int main(void)
{
  mmapGpio rpiGpio;
  //Set outputs
  rpiGpio.setPinDir(PINA0,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PINA1,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PINA2,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PINA3,mmapGpio::OUTPUT);
  //TURN OFF ASAP!
  rpiGpio.setPinDir(PINCS0,mmapGpio::OUTPUT);
  rpiGpio.writePinHigh(PINCS0);
  //TURN OFF ASAP!
  rpiGpio.setPinDir(PINCS1,mmapGpio::OUTPUT);
  rpiGpio.writePinHigh(PINCS1);
  rpiGpio.setPinDir(PIND0,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PIND1,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PIND2,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PIND3,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PIND4,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PIND5,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PIND6,mmapGpio::OUTPUT);
  rpiGpio.setPinDir(PIND7,mmapGpio::OUTPUT);

unsigned char dataStore[2][16];
```

The Code LED MATRIX

13

```
for(int j=0;j<2;j++)
{
for(int k=0;k<16;k++)
{
  dataStore[j][k]=64;
}
}
```

09

```
  int blockCounter=0;
  int rowCounter=0;
      while(1)
    {
    blockCounter++;
    if(blockCounter==16)
    {
      if(rowCounter==0)
      {
        blockCounter=0;
        rowCounter=1;
      }
      else
      {
        blockCounter=0;
        rowCounter=0;
      }
    }
```

10

```
    safelySetRow(rowCounter, &rpiGpio);
    setAddress(blockCounter, &rpiGpio);
    setData(dataStore[rowCounter][blockCounter], &rpiGpio);
    usleep(50);
    }
  return 0;
}
```

FAQ What is OpenElec?

Kodi fans might have heard of this Linux distribution.
What makes it so special?

What is OpenELEC, then?

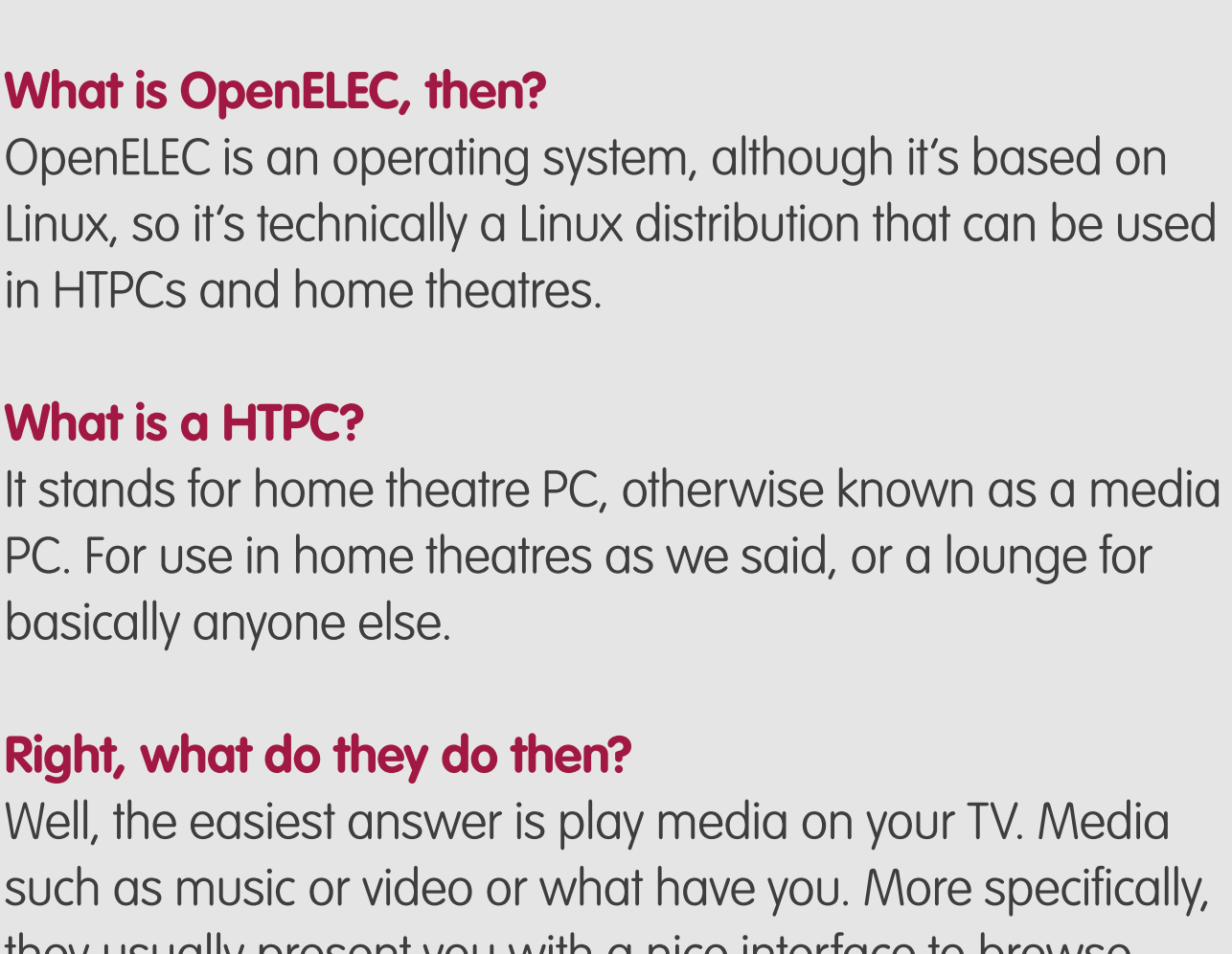
OpenELEC is an operating system, although it's based on Linux, so it's technically a Linux distribution that can be used in HTPCs and home theatres.

What is a HTPC?

It stands for home theatre PC, otherwise known as a media PC. For use in home theatres as we said, or a lounge for basically anyone else.

Right, what do they do then?

Well, the easiest answer is play media on your TV. Media such as music or video or what have you. More specifically, they usually present you with a nice interface to browse your available media on, either stored locally on a hard drive, on USB storage or over the network on another computer. It then plays that media with all the features of a full PC video player, such as VLC.



> "They usually present you with a nice interface to browse your available media on, either stored locally on a hard drive, on USB storage or over the network"

Left OpenELEC is accessible straight from the official downloads page

What's the benefit of doing this?

Well you can then watch this stuff on your TV rather than at your computer. Some people find it more comfortable and it can be a bit more social.

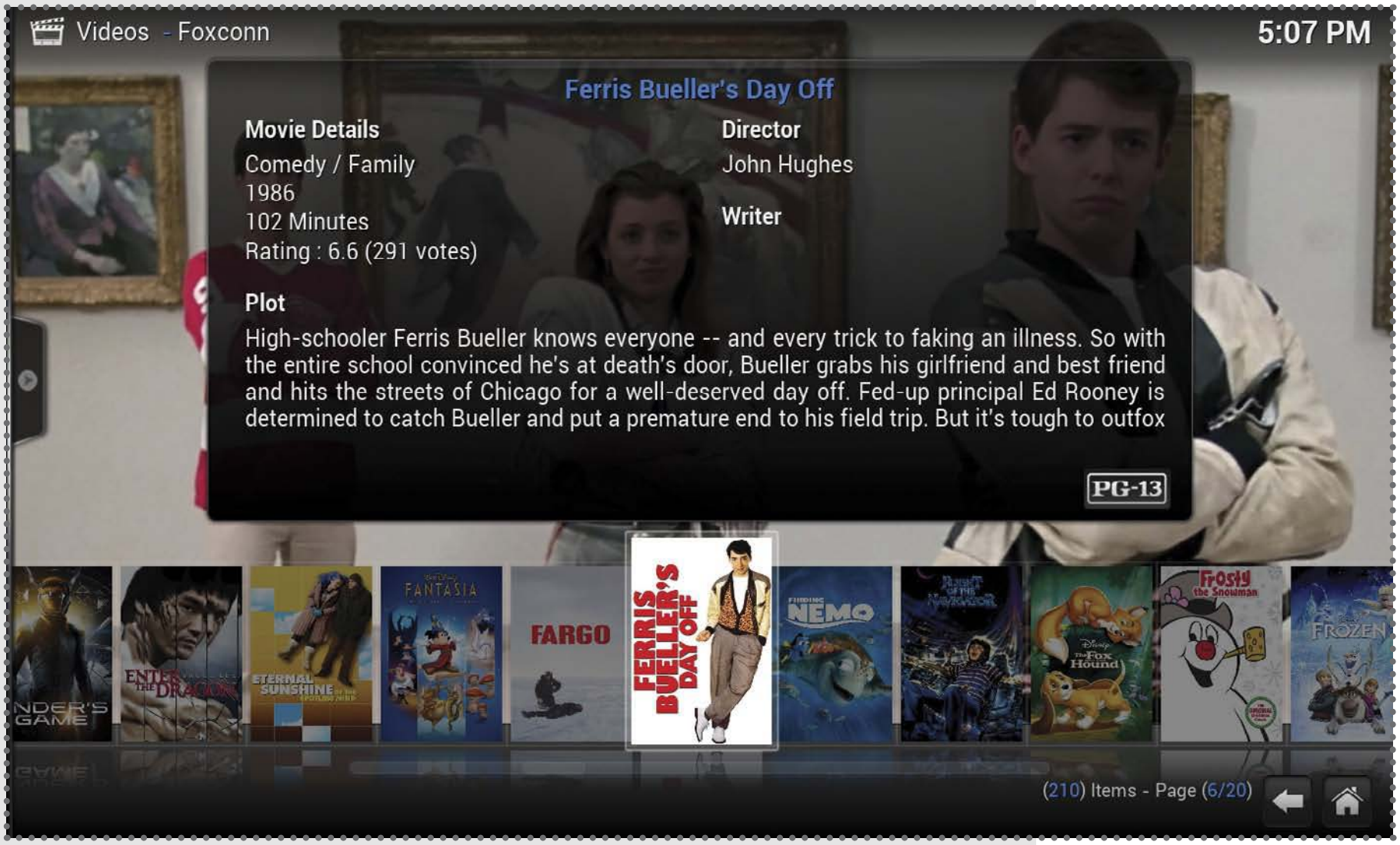
Oh, that makes sense. What exactly is the significance of OpenELEC in all of this then?

OpenELEC is one of the premier ways of getting the Kodi software onto a device, such as a computer connected to your TV.

What is Kodi?

The new name for XBMC, which used to be the Xbox Media Centre a lifetime ago when hacking an original Xbox was a thing that people did. It's HTPC software and very popular at that, with an easy-to-use interface and a codebase used by some other popular HTPC software.

Below Kodi resembles streaming platforms like Netflix, or those on the Xbox and PS4



Left For €90 you can get a dedicated OpenELEC box – go to **http://bit.ly/1SBhAY3**

Can I get Kodi anywhere else?

Yep, it's available on a lot of Linux distros and other operating systems, and there are a handful of dedicated ones that run Kodi for HTPCs.

Okay, so why is OpenELEC important?

OpenELEC has ties to the Kodi development team – in fact some of the Kodi devs also work on OpenELEC. It also supports a huge range of different devices, including your average PC towers and laptops, to Apple TVs and other TV hardware along with many other things too.

I'm guessing that the Pi is included in this as well?

Absolutely, in fact they make a big deal out of whenever an OpenELEC release does something cool on the Raspberry Pi – the Raspberry Pi has become extremely popular as a HTPC due to its size and power consumption. You can even attach it to the back of some TVs.

How can I get it for Pi?

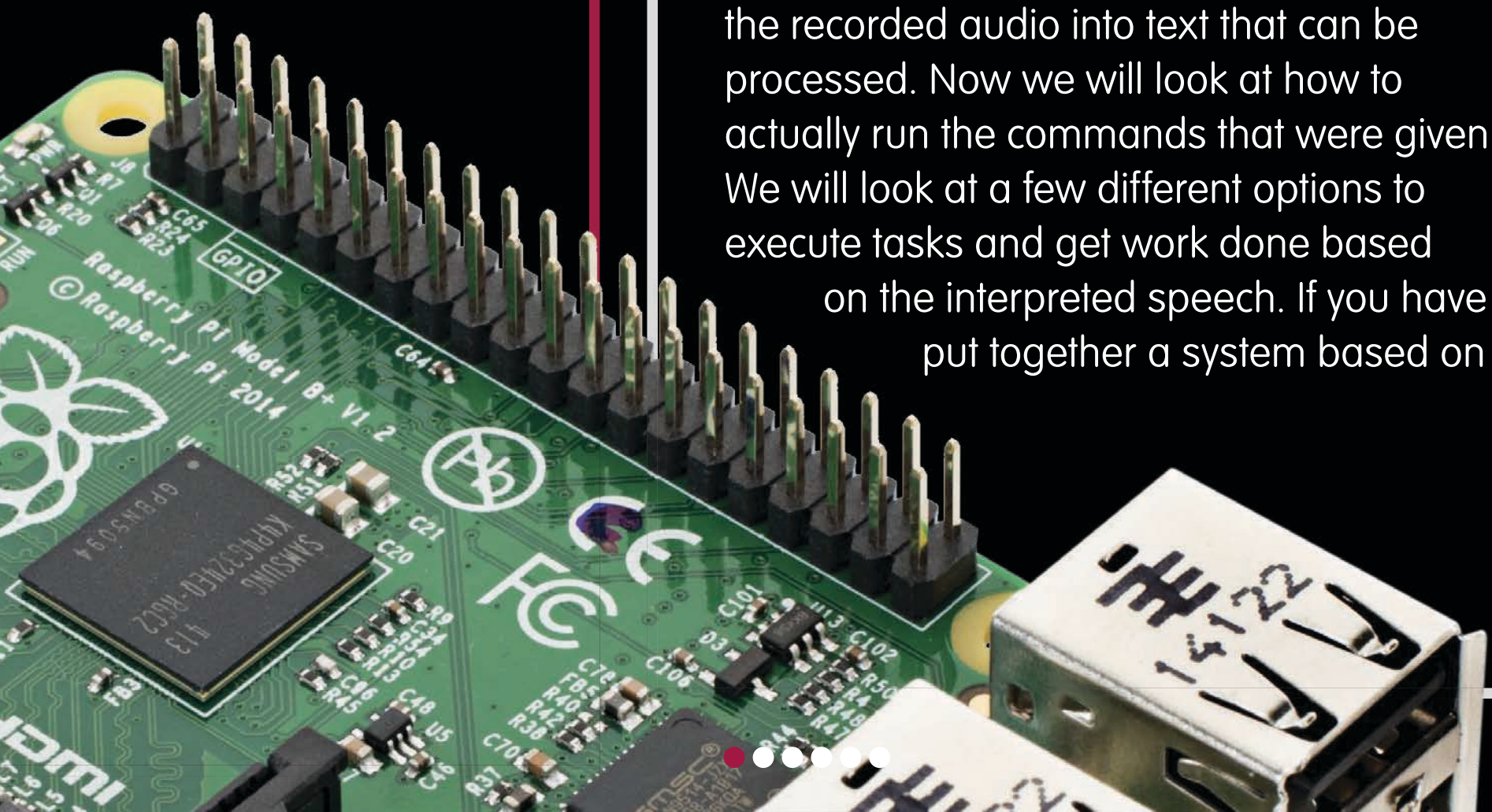
Just head to **https://www.raspberrypi.org/downloads** and then scroll down to find the download link.

Digital assistant, part 3: running other programs

This final article in our J.A.R.V.I.S. series will cover how to actually run the commands you are giving to your Pi

"We will look at a few different options to execute tasks and get work done based on the interpreted speech"

This is the last in our trilogy of articles to help you build your own voice control system. The first article, in **RasPi** #17, looked at how to listen for incoming commands. This involved listening on a USB device and also outputting audio feedback to a user. The second article looked at how to interpret those commands. This involved using speech recognition libraries to translate the recorded audio into text that can be processed. Now we will look at how to actually run the commands that were given. We will look at a few different options to execute tasks and get work done based on the interpreted speech. If you have put together a system based on



the suggestions from the first two articles, you should have a string containing the text that was spoken to your Raspberry Pi. But, you need to figure out what command this maps to. One method is to do a search for keywords. If you have a list of keywords available, you can loop through them and search the heard string to see if any one of those keywords exist within it as a substring. Then you can execute the associated task with that keyword. However, this method will only find the first match. What happens if your user accidentally includes a keyword in their spoken command before the actual command word? This is the auditory equivalent to having fat fingers and mistyping a command on the keyboard. Being able to deal with these errors gracefully is an ongoing area of research. Maybe you can create a new algorithm to handle these situations?

Let's say that you have a series of Python scripts that contain the various tasks you want your system to be able to tackle. You need a way to have your system be able to run these scripts when called upon. The most direct way to run a script is to use execfile. Say that you have a script called do_task.py that contains Python code that you want to run when a command is given; you can run it with:

```
execfile("do_task.py")
```

Using this form, you can add command line options to the string being handed in. This will look in the current directory for the script of that file name and run it in the current execution context of your main program. If you need to rerun this code multiple times, call execfile each time you do. If you don't need the script to run within the same context, use the subprocess module. Import it with:

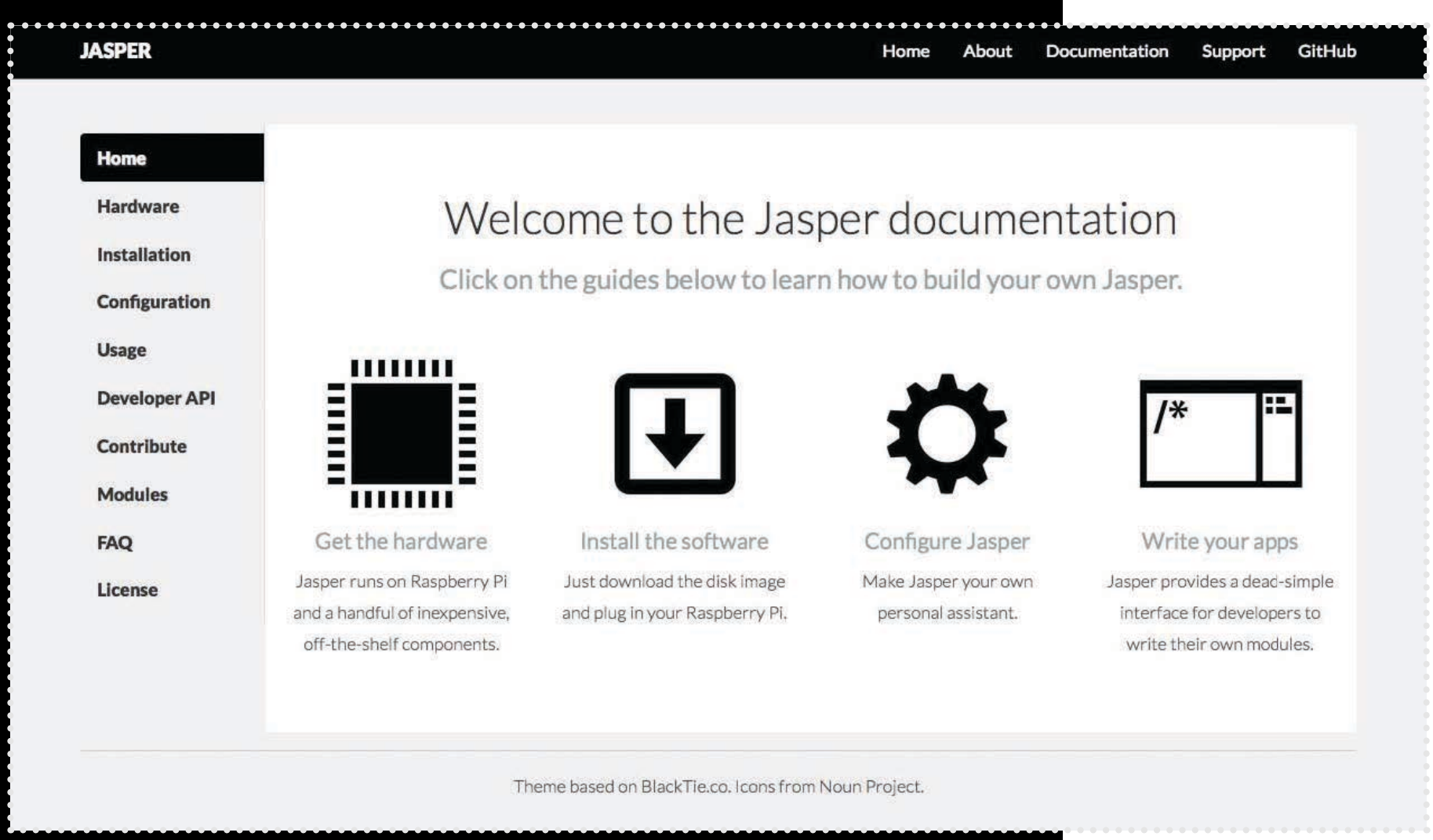
"You should have a string containing the text that was spoken to your Raspberry Pi. But, you need to figure out what command this maps to"

```
import subprocess
```

You can then execute the script like so:

```
subprocess.call("do_task.py")
```

This will fork off a subprocess of the main Python interpreter and run the script there. If your script needs to interact with the main program, this is probably not the method that you should use. Collecting output from a call to do_task.py with subprocess isn't straightforward, so another way of achieving the same thing is to use the import statement. It also runs the code in your script at the point the import statement is called. If your script only contains executable Python statements, these get run at the point of importation. In order to rerun this code, you need to use the reload command. The reload command doesn't exist in version three – so if you're



Below The Jasper project has some great documentation that might help guide you further in terms of hardware and software choices

using that particular Python version, a better option is to encapsulate the code contained in the script within a function. You can then import the script at the beginning of your main program and simply call the relevant function at the correct time. This is a much more Pythonic method to use. If you have the following contents for do_task.py:

```
def do_func():
  do_task1()
  do_task2()
```

You can then use it with the following code within your main program:

```
import do_task
....
....
do_task.do_func()
....
```

An even more Pythonic method is to use classes and objects. You can write a script that defines a class that contains methods for you to call when you need it.

What are the options if you want to do something that isn't achievable with a Python script? In these cases, you need to be able to run arbitrary programs on the host system. The host system in this case is your Raspberry Pi. As a toy example, let us say you need to download some emails using the Fetchmail program. You can do this in a couple of different ways. The older method is to use the os.system() command where you hand in a string. In our example, this would look something like the following:

Social media

There are several Python modules available for checking social media accounts. For example, run:

```
sudo apt-get
install python-
facebook
```

You can then use import facebook to access the Facebook API. For the Twitter API, install the python-twitter Debian package. Email is easier as long as your email provider offers IMAP or POP access. Google has a Python module that gives access to the APIs for your calendar, email and fitness data.

```
os.system("/usr/bin/fetchmail")
```

You need to explicitly use os.wait() to be told exactly when the task has finished. This method is now being replaced by the newer subprocess module. It gives you more control over how the task gets run and how you can interact with it. A simple equivalent to the above command would look like this:

```
subprocess.call("/usr/bin/fetchmail")
```

It waits until the called program has finished and returns the return code to your main Python process. But what if your external program needs to feed in results to your main program? In this case, you can use the command: subprocess.check_output(). This is essentially the same as subprocess.call(), except when it finishes, anything written out by the external program to stdout gets handed in as a string object. If you also need information written out on stderr, you can add the parameter stderr=subprocess.STDOUT to your call to subprocess.check_output.

After reading these three articles, you should have enough of the bare bones to be able to build your own version of the J.A.R.V.I.S system. You will be able to fine-tune it to do basically anything that you command it to do. So go forth and order your machines around as you see fit, and have them actually listen to what you are saying for once!

"What if your external program needs to feed in results to your main program? In this case, you can use the command: subprocess.check_output()"

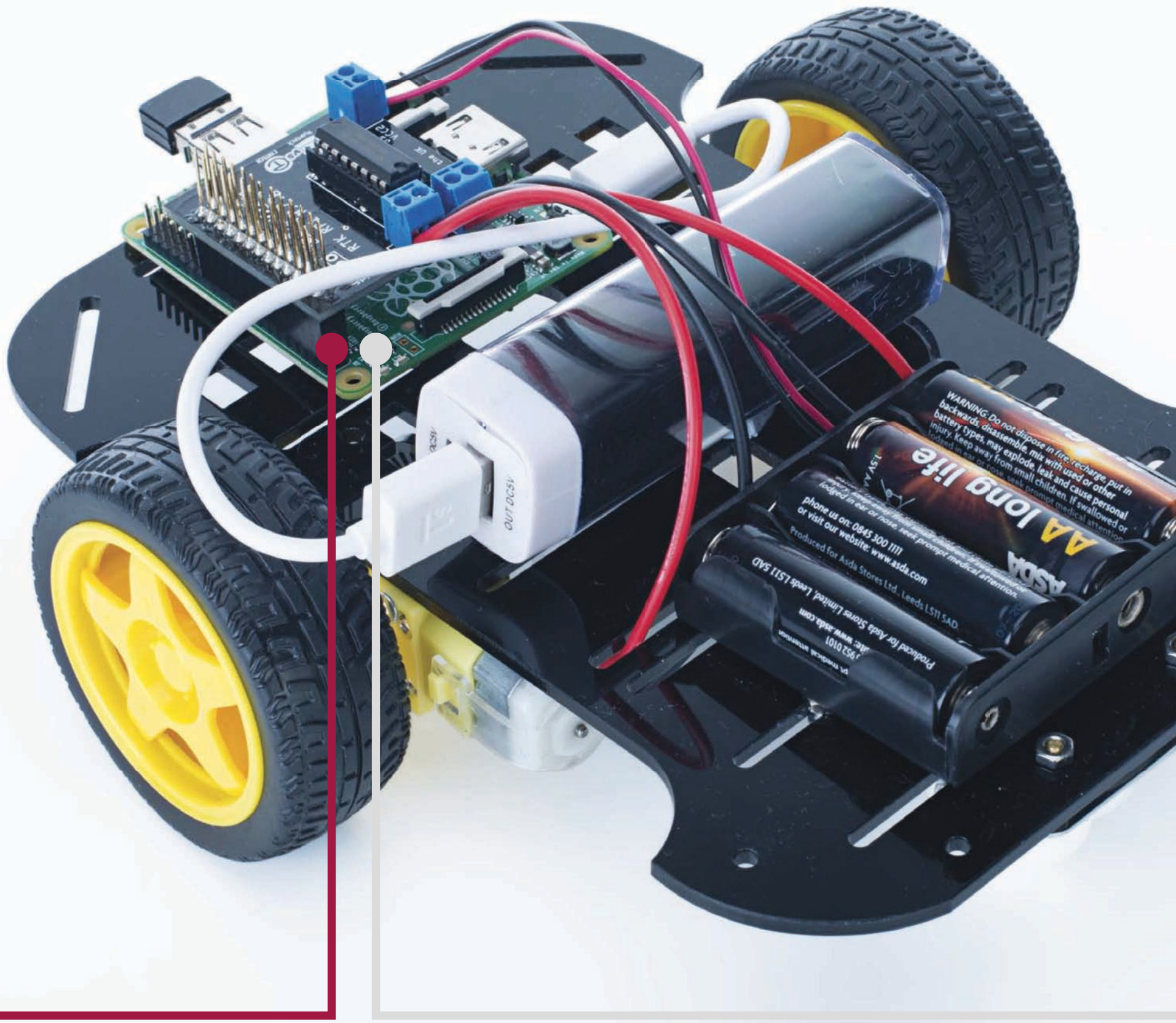
The Code DIGITAL ASSISTANT

do_task.py
```
def do_func():
   print “Hello World”
```

main_program.py
```
# You can import your own module to do tasks and commands
import do_task

# You can then go ahead and run any included functions
do_task.do_func()

# You can run system programs directly
import os

# The exit code from your program is in the variable returncode
returncode = os.system(“/usr/bin/fetchmail”)

# The subprocess module is a better choice
import subprocess

# You can duplicate the above with
returncode = subprocess.call(“/usr/bin/fetchmail”)

# If you want to get the output, too, you can use
returned_data = subprocess.check_output(“/usr/bin/fetchmail”)
```

Ryanteck Budget Robotics Kit for Raspberry Pi

The cheapest and simplest Raspberry Pi robot you're going to find on the market is a perfect entry-level kit



We have had some low-price and basic kits in the office over the past few months, each with varying levels of success. Some have just been cheaply made, others have used a simple selection of components that enable expansion when you feel more comfortable with the robotics behind it. However, the Ryanteck budget kit that we're reviewing here takes a slightly different approach.

The budget in its name is definitely a very apt description. The kit comes with the bare essentials needed to get locomotion: two motors, a chassis to attach them to and a way to connect them to a Raspberry Pi. You can spend a little more to get the whole thing ready to build when you get the kit, but otherwise you'll also need a Raspberry Pi, a Wi-Fi dongle and a portable battery for powering the Pi.

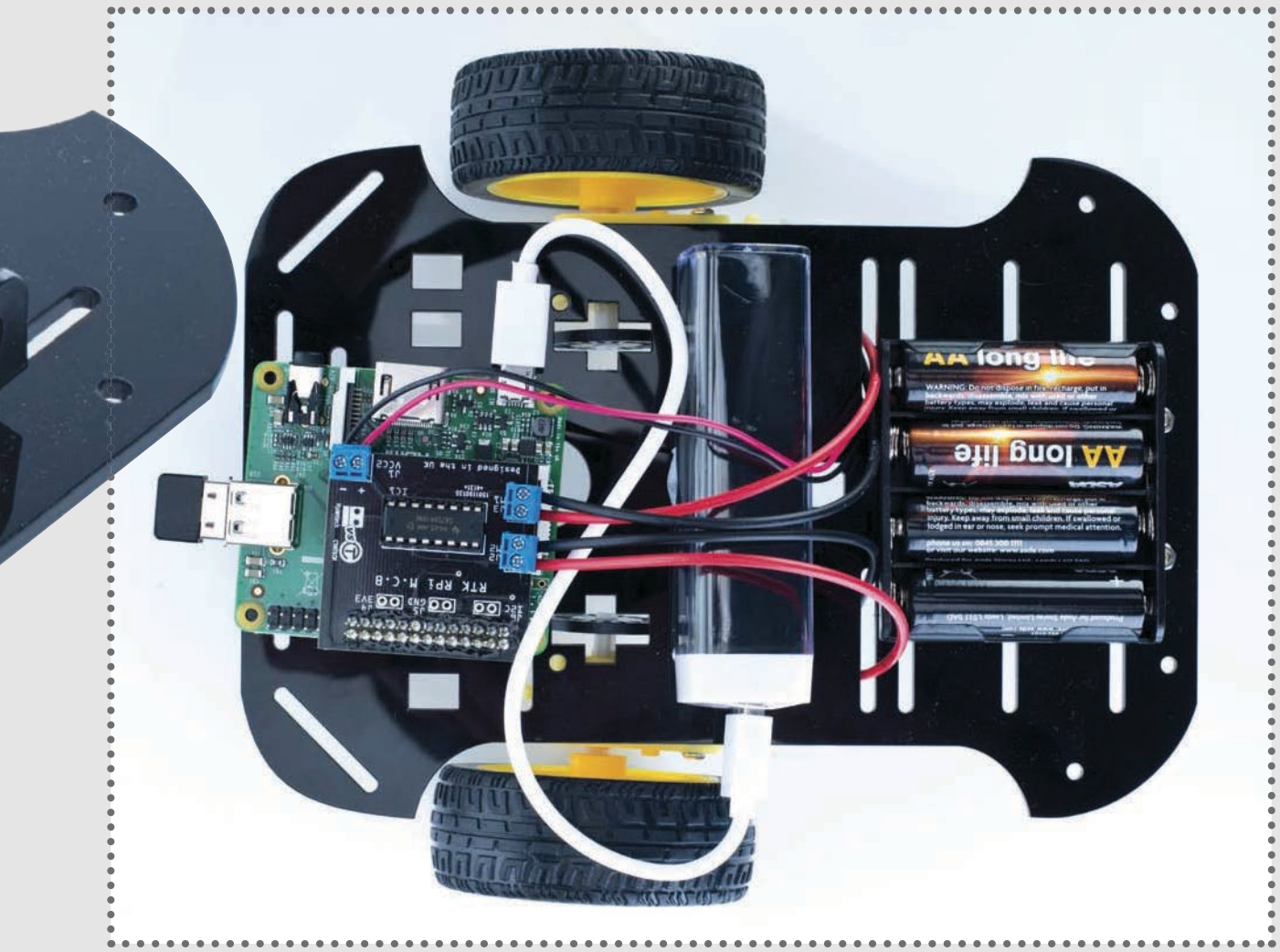
It's quite quick and simple to put together and it's fairly robust to boot. The motors are the standard ones

TECH SPECS

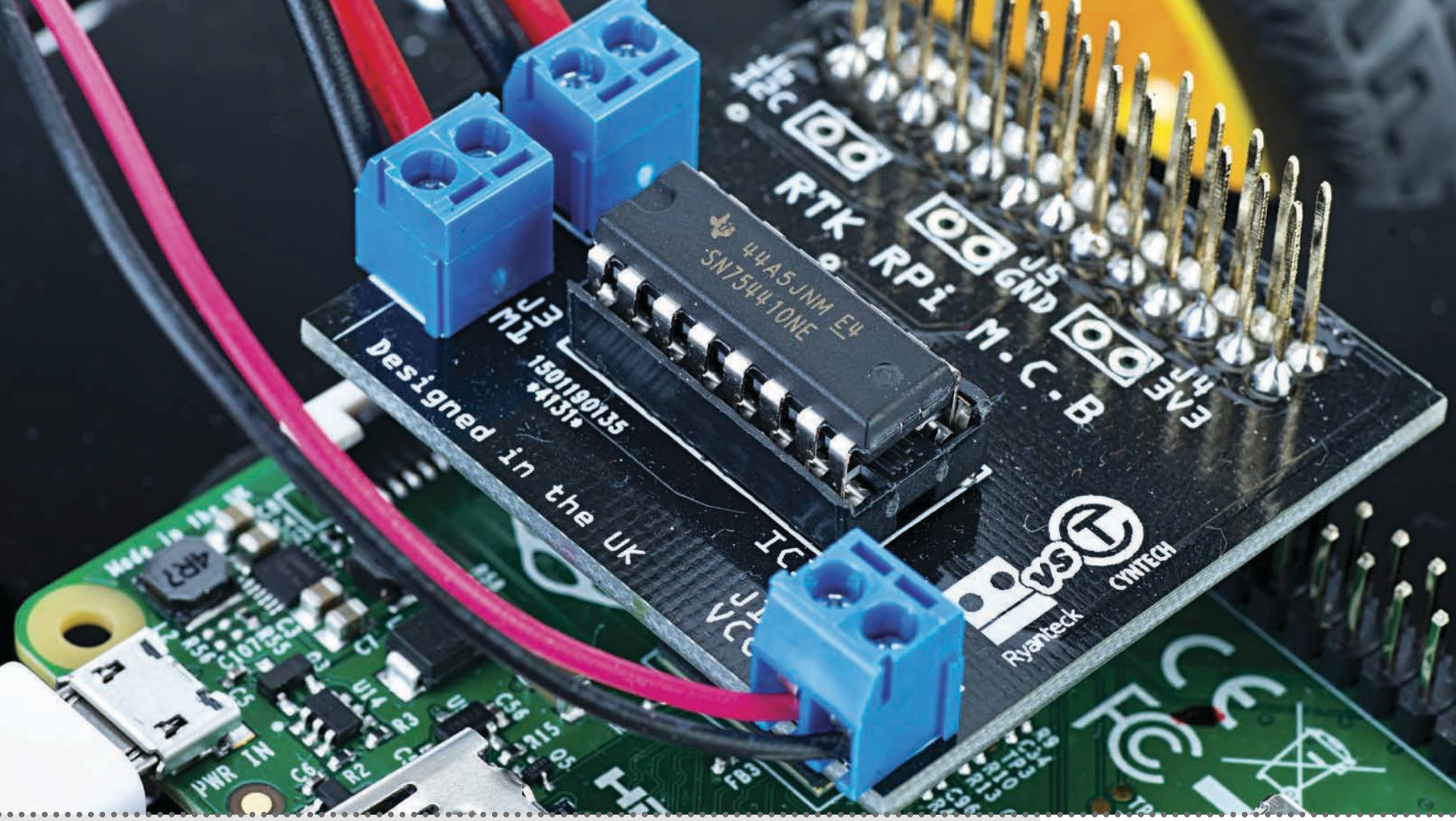
Dimensions
20 x 170 x 80mm

Weight
0.5 kg (including battery)

Price
£24.49



Left The chassis is roomy enough to include a portable power pack and a few different HATs



you'll get on a lot of other robot kits and they even have a speed sensor if you want to add the ability to better control straight-line performance. The chassis itself has plenty of mounting points to add any number of extra sensors with a little re-jig of the components on top.

Programming the robot is fairly simple, using Python and the GPIO ports to activate the motors as you see fit. There's no Ardunio to worry about and the basic example code on the website is good enough to adapt for your own uses. With the way the board attaches to the Raspberry Pi, the standard 26 pins are available to plug into it – so you can add sensors that way with a bit of ingenuity.

It is a simple kit that doesn't look like much, but for the price and the ability to get started with robotics pretty much straight away, it's a great package. Its upgrade paths are very limited though, so you'll soon need to start thinking about other robots if you want to do more.

Above The kit has a Ryanteck RPi Motor Controller Board

THE VERDICT

Very good for complete novices, kids and party pieces with your Raspberry Pi, but it's the barest-bones entry level kit for robotics that is more difficult to expand upon compared to other budget kits. None of them are as cheap as this though, or work so readily with a Model A+

Next issue

Get inspired · Expert advice · Easy-to-follow guides

Build a Minecraft Console

Plus Deploy special moves with a Minecraft power glove

Get this issue's source code at:
www.linuxuser.co.uk/raspicode